



IES El Palau

Treball de Recerca

DOMÒTICA

Jia Long Ji Qiu

Dirigit per Carles Amigó

2n de Batxillerat, B

Sant Andreu de la Barca, 5 de desembre de 2018

ÍNDEX

0. INTRODUCCIÓ	3
1. BLOC TEÒRIC	4
1.1. Domòtica	4
1.1.1. Elements de la domòtica	4
1.1.2. Funcionament	5
1.1.3. Classificació dels sistemes domòtics	6
1.2. Hardware	8
1.2.1. Tipus de hardware	8
1.2.2. Arduino UNO Rev3	9
1.2.3. Mòdul Bluetooth HC-06	12
1.2.4. Sensors	14
1.2.5. Actuadors	17
1.3. Software	19
1.3.1. Tipus de software	19
1.3.2. Què és programar?	20
1.3.3. Arduino IDE	21
1.3.4. App Inventor	22
1.3.5. Fritzing	23
2. BLOC PRÀCTIC	24
2.1. Planificació de la maqueta	24
2.1.1. Plànols	24
2.1.2. Circuit elèctric	25
2.2. Programació	29
2.2.1. Programa d'Arduino	29
2.2.2. Aplicació mòbil	35
3. CONCLUSIONS	41
4. WEBGRAFIA	42
5. ANNEXOS	45
5.1. Annex 1, programa de l'Arduino	45
5.2. Annex 2, programa de l'aplicació mòbil	48

0. INTRODUCCIÓ

L'ésser humà porta depenent de la tecnologia des dels seus orígens. Els prehistòrics, per exemple, ja empraven diverses tècniques per fabricar armes i eines per tal de poder sobreviure. Però és gràcies a la ciència, a l'empeny que tenim per comprendre l'univers, el que realment va afavorir l'avanç de la tecnologia fins arribar a com és actualment. Podem afirmar, doncs, que la tecnologia, acompanyat amb la ciència, són essencials pel desenvolupament de la nostra espècie.

Està clar que estem molt familiaritzats amb la tecnologia, però, què és exactament? Per definir aquest concepte cal tenir en compte el seu origen etimològic. Aquest terme prové del grec, format per *tekne* (tècnica, ofici) i *logos* (ciència, coneixement). Dit això, podríem definir-ho com el conjunt de coneixements relacionats amb els oficis, procediments i tècniques artesanals o industrials, destinats a la fabricació d'eines i objectes, amb l'objectiu de modificar el món per adaptar-lo a les necessitats humanes.

Actualment, a la vida quotidiana, utilitzem la tecnologia sobretot per afavorir el benestar de la vida diària de les persones. Aquest afany ha originat moltes tècniques noves, i una d'aquestes és la domòtica.

En aquest projecte explicaré més detalladament com funciona exactament la domòtica i alguns dels seus mecanismes. A més a més, es recrearà un petit sistema domòtic dins d'una maqueta. Per tant, aquest document constarà principalment d'una part teòrica i una altra pràctica.

El meu objectiu principal és dissenyar i construir una maqueta d'un habitatge domèstic amb un sistema domòtic instal·lat, aconseguint que tot l'habitatge pugui ser automatitzat o controlat per una aplicació mòbil, tot això partint d'uns coneixements mínims d'electricitat i de programació. Degut a aquesta manca d'informació, m'agradaria també aprendre més sobre programació, Arduino i els circuits elèctrics durant la elaboració d'aquest treball.

1. BLOC TEÒRIC

1.1. Domòtica

La domòtica és un conjunt de tecnologies aplicades al control i automatització intel·ligent de qualsevol tipus d'edificació, principalment d'habitatges.

L'automatització de certes funcions tenen les seves pròpies finalitats. Dins d'aquestes, la que més predomina és el confort, que s'aconsegueix mitjançant la realització automàtica de certes tasques repetitives. També hi ha d'altres bastant importants com l'estalvi energètic i la seguretat.

1.1.1. Elements de la domòtica

Els elements principals que componen la instal·lació d'un sistema domòtic són un controlador, una interfície d'usuari, uns sensors i uns actuadors.

- Controlador domòtic

En un passat ja utilitzàvem diversos sistemes de control, els quals gestionaven la calefacció, la llum, etc., però, gràcies a l'avenç tecnològic de la domòtica, tots aquest sistemes s'han pogut centralitzar en un únic dispositiu, el controlador domòtic.

Podríem dir que el controlador domòtic, també conegut com el nucli del sistema, es un “cervell electrònic”, perquè de la mateixa manera que els essers humans captem informació mitjançant els nostres sentits, l'enviem al cervell i fem certes accions, el controlador és l'element que s'encarrega de rebre i processar tota la informació obtinguda pels sensors i, posteriorment, envia diferents ordres a cadascun dels actuadors que s'encarreguen d'executar-los.

- Interfície d'usuari

La interfície d'usuari és el mitjà amb el qual l'usuari es pot comunicar amb una màquina, facilitant d'aquesta manera el control i el funcionament d'aquesta. Simultàniament, la màquina també pot enviar informació a l'usuari, anomenat *feedback*, de manera que es pugui saber si una acció ha funcionat correctament.

- Sensors

Els sensors són dispositius formats per cèl·lules sensibles que, quan detecten variacions en uns paràmetres externs (la temperatura, el vent, la llum, accions humanes...), les tradueixen en senyals elèctriques i les transmeten a un controlador domòtic.

- Actuadors

Els actuadors fan la funció inversa dels sensors, són dispositius que converteixen una senyal elèctrica en una determinada acció, sol·licitada pel controlador domòtic. D'aquesta manera, es canvien els paràmetres externs i es dona lloc a una situació concreta, programada prèviament.

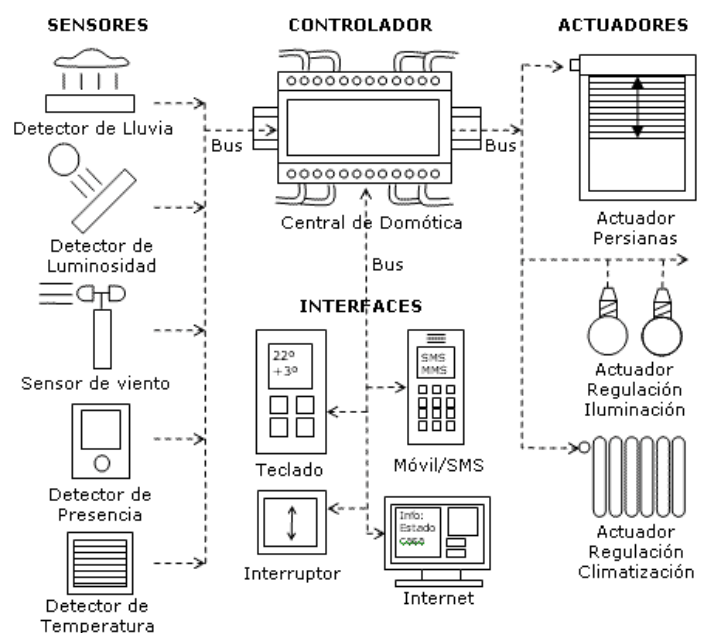
- Bus de control

El bus de control és l'element que permet les interconnexions elèctriques entre tots els elements prèviament anomenats. La transmissió més habitual de senyals es realitza a través de cables, però també hi ha un altre mètode que emprava com a medi de transmissió l'aire.

1.1.2. Funcionament

El funcionament d'un sistema domòtic és bastant senzill. Com ja s'ha explicat abans, el sistema domòtic consta de sensors, actuadors, interfícies, controladors i busos de control, que interconnecten tots aquests elements.

Quan un sensor detecta en el medi extern una variació en cert paràmetre, aquest ho tradueix en senyals elèctriques que s'envien al controlador domòtic. El controlador processa tota aquesta informació i, acte seguit, envia uns ordres determinats a l'actuador corresponent, que realitzarà l'acció sol·licitada per tal de donar lloc a la situació desitjada per l'usuari. A més a més, certes funcions també es poden controlar manualment, a través d'una interfície d'usuari.



IL·LUSTRACIÓ 1. SISTEMA DOMÒTIC

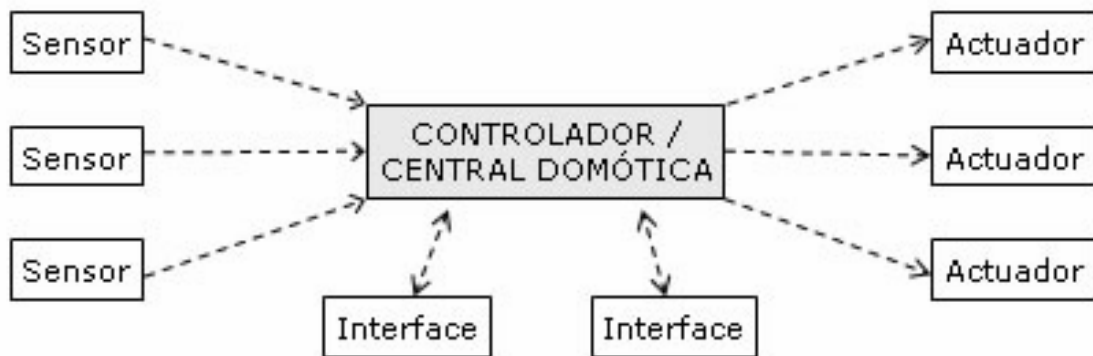
Per acabar d'entendre aquesta explicació, posaré com a exemple una alarma: un sensor de presència detecta una intrusió, envia informació al controlador dient-li que ha detectat a algú, ho processa i envia com a ordre que s'activi un bronzidor, l'actuador. Lògicament, aquesta alarma s'activarà sempre i quan ho vulgui l'usuari, i per això es farà servir una interfície, on l'usuari podrà "dir-li" al controlador que activi l'alarma, un cop estigui fora de l'habitatge.

1.1.3. Classificació dels sistemes domòtics

Segons l'arquitectura d'un sistema domòtic, és a dir, la forma en la que està estructurada la xarxa, els classifiquem en sistemes centralitzats, descentralitzats o distribuïts.

- Sistema centralitzat

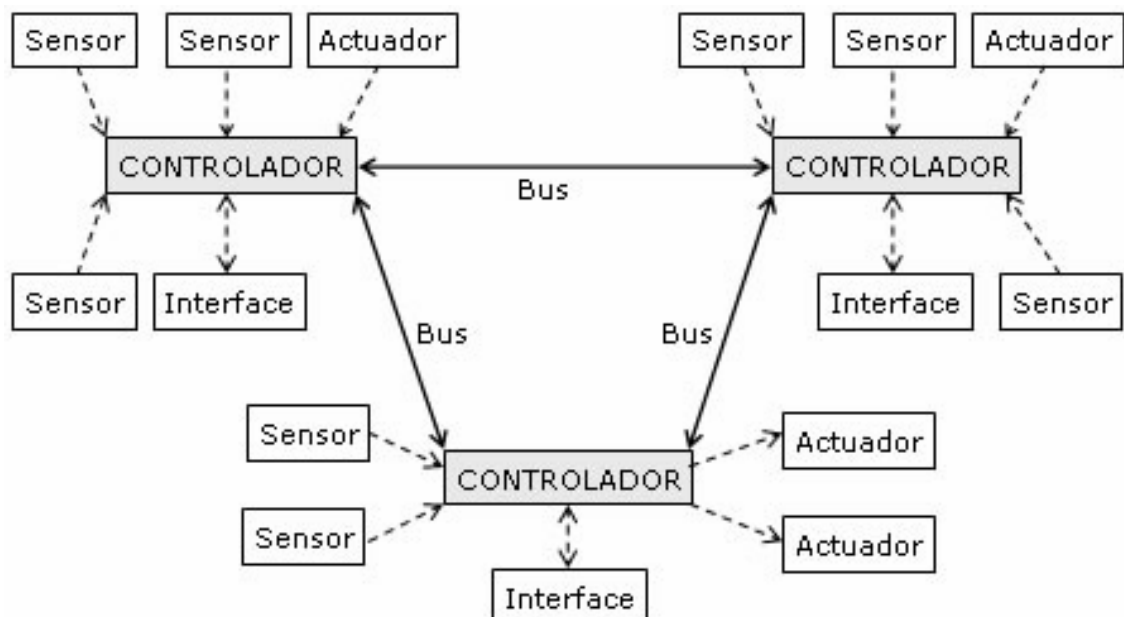
Els elements que formen un sistema centralitzat depenen d'un element principal, el controlador, que s'encarrega de rebre informació dels sensors i enviar instruccions als actuadors.



IL·LUSTRACIÓ 2. SISTEMA CENTRALITZAT

- Sistema descentralitzat

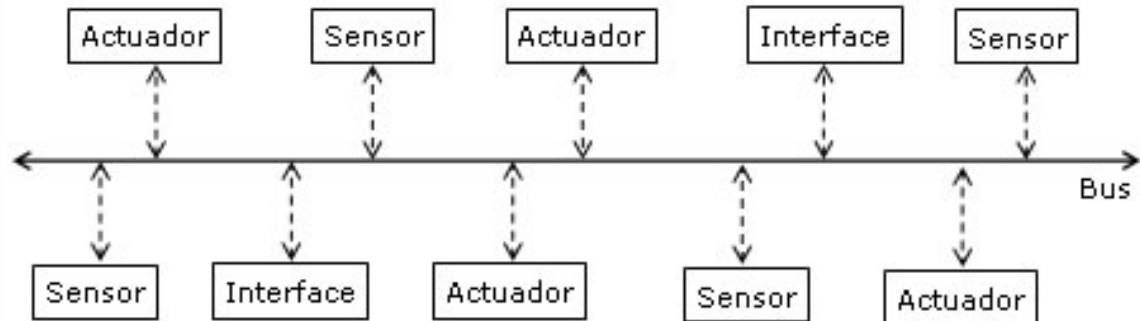
En aquest tipus d'arquitectura els sensors i actuadors estan connectats a diferents controladors, tots aquests interconnectats mitjançant un sistema de bus de control la qual permet la comunicació entre ells.



IL·LUSTRACIÓ 3. SISTEMA DESCENTRALITZAT

- **Sistema distribuït**

Els elements d'un sistema distribuït no depenen d'un element central, sinó que cada sensor i actuador actuen com controladors i poden funcionar per si mateixos.



IL·LUSTRACIÓ 4. SISTEMA DISTRIBUÏT

També es poden classificar segons el protocol de comunicació; el “llenguatge”, o millor dit, el conjunt de normes que utilitzen els dispositius que formen un sistema domòtic per comunicar-se. Distingim entre dos tipus, tancat (de propietari) o obert (estàndard).

- **Protocols tancats o de propietari**

Els protocols tancats són específics d'una marca en particular, on només el fabricant pot manipular els dispositius.

- **Protocols oberts o estàndards**

Els protocols oberts, en canvi, són totalment lliures, de manera que qualsevol fabricant pot utilitzar el mateix “llenguatge”, amb la finalitat unificar el sistema independentment de les marques.

Per últim, tenim la classificació segons el medi de transmissió de dades, el qual fa referència al suport físic a través del qual un emissor i receptor poden comunicar-se. Aquests poden ser guiats o no guiats. En tots dos casos, la transmissió s'empra mitjançant ones electromagnètiques.

- **Transmissió guiada (amb cable)**

Les ones són transmeses a través d'un camí físic, els cables, que s'encarreguen de la conducció d'un extrem a altre.

- **Transmissió no guiada (sense cable)**

Aquest tipus de transmissió és permès gràcies a uns suports físics que la faciliten, però les ones no són dirigides.

1.2. Hardware

Tot i que ja sabem com funciona exactament la domòtica, abans de començar amb la part pràctica, cal entendre cadascun dels elements físics que farem servir en aquest projecte, és a dir, el hardware.

Entenem el hardware com la part física i tangible de la informàtica, allò que s'utilitza perquè qualsevol sistema informàtic, no necessàriament un ordinador, pugui funcionar i executar correctament les tasques que se li programen.

1.2.1. Tipus de hardware

Principalment, podem classificar el hardware en dues categories. D'una banda tenim el bàsic, que engloba tots aquells components imprescindibles perquè el sistema informàtic funcioni correctament. D'altra, tenim el complementari que, com indica el seu nom, fa referència a tots aquells dispositius addicionals, no estrictament necessaris, que serveixen per realitzar funcions específiques per tal de completar o millorar un sistema informàtic. Però, aquests dispositius també es poden classificar segons les seves funcions.

- **Hardware de processament**

El hardware de processament té la finalitat d'interpretar i executar diferents instruccions.

- **Hardware de emmagatzematge**

Aquests tipus de dispositius tenen la capacitat d'emmagatzemar dades i informació, ja sigui de manera temporal o permanent.

- **Hardware d'entrada**

Els dispositius d'entrada permeten la introducció de dades i informació, des d'un medi exterior, a un sistema informàtic.

- **Hardware de sortida**

Contràriament als d'entrada, aquests són capaços de dirigir les dades generades pel sistema informàtic cap a l'exterior.

- **Hardware bidireccional (d'entrada i sortida)**

Es tracta de dispositius que combinen les funcions dels d'entrada i de sortida, és a dir, aquest tipus de hardware té la capacitat de permetre la introducció de dades i, simultàniament, dirigir-les cap a l'exterior.

Un cop explicat el hardware, ja podem procedir a l'explicació dels elements que farà servir per a la meua maqueta.

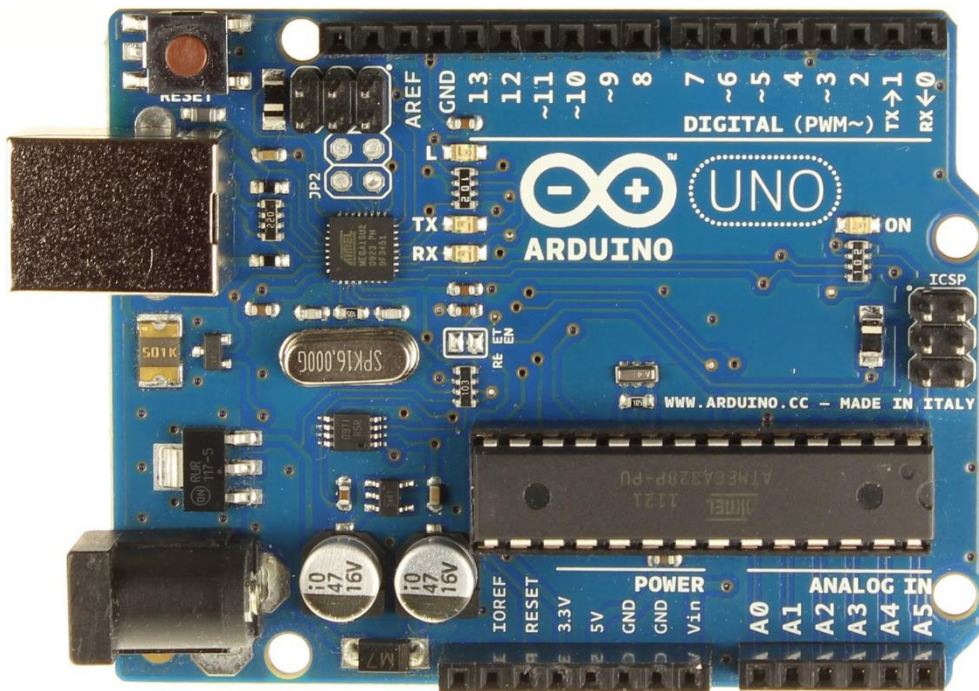
1.2.2. Arduino UNO Rev3

Arduino és una plataforma electrònica oberta tant de hardware com de software dedicada a la creació de prototips, cosa que permet qualsevol modificació per part de l'usuari. Gràcies a això i al poc coneixement que requereix i a la seva senzillesa a l'hora d'utilitzar-la, gairebé qualsevol persona interessada en la electrònica i la informàtica podria dissenyar i crear el seu propi projecte. Aquests aspectes són justament els que han fet que triés Arduino per realitzar aquest treball.

Tot i que el catàleg que ofereix Arduino és molt variat, en aquest projecte em centraré en la placa Arduino UNO Rev3. Aquesta placa es tracta d'un microcontrolador, basat en l'ATmega328P, connectat amb un circuit intern molt complex. Els elements més rellevants que formen aquesta placa són els pins d'entrada i/o sortida, la unitat central de processament (CPU), els diferents tipus de memòria, un connector d'alimentació i un altre d'USB.

ESPECIFICACIONS TÈCNIQUES DE LA PLACA ARDUINO UNO REV3

Voltatge d'operació	5V
Voltatge d'entrada (recomanada)	7-12V
Voltatge d'entrada (límit)	6-20V
Corrent contínua per pin d'E/S	20 mA
Corrent contínua per pin de 3.3V	50 mA
Memòria flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocitat de rellotge	16 MHz
Dimensions	53.4 mm x 68.6 mm
Pes	25 g



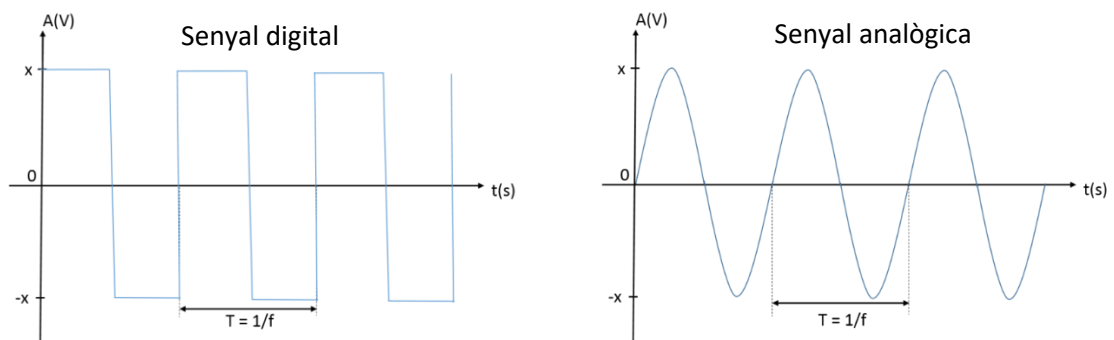
IL·LUSTRACIÓ 5. PLACA ARDUINO UNO REV3

- Pins d'entrada i/o sortida i UCP

Aquesta placa disposa de 14 pins digitals d'entrada i sortida, dels quals 6 poden ser utilitzats com sortides PWM (*Pulse-Width Modulation*), i 6 pins analògics d'entrada. Abans de procedir a explicar cada tipus de pin, cal clarificar què són les senyals digitals, analògiques i PWM.

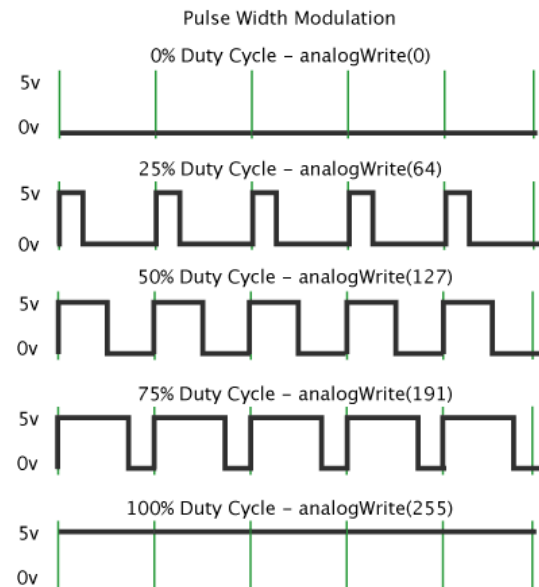
Una senyal digital és una variació de voltatge discontinua entre el valor mínim del voltatge, 0V, i el valor màxim, 5V (en el cas de l'Arduino), que no té en compte els valors intermedis, per tant, només té dos estats possibles, un nivell alt (5V) o baix (0V).

La senyal analògica, en canvi, té una variació contínua i permet utilitzar qualsevol valor que es troba dins l'interval del valor mínim i màxim, gràcies a això, podem obtenir valors analògics infinits, d'entre 0V i 5V.



IL·LUSTRACIÓ 6. GRÀFIQUES DE LES SENYALS DIGITAL I ANALÒGICA

Ja que l'Arduino no ofereix una sortida analògica, utilitza una senyal modulada per amplada de pols (PWM). Es tracta d'una senyal digital que permet obtenir resultats analògics, com si s'enviés una senyal analògica. Aquest efecte s'aconsegueix a partir d'una variació del cicle de treball d'una senyal digital, és a dir, dins d'un cicle de treball s'activa la sortida digital només durant un cert període de temps, de manera que el voltatge mitjà obtingut durant un cicle sencer serà igual al valor analògic desitjat. Per exemple, per obtenir 2.5V cal utilitzar el 50% d'un cicle de treball.



IL·LUSTRACIÓ 7. GRÀFIQUES DE SENYAL PWM

Els pins digitals poden ser configurats com entrades o sortides. Els d'entrada s'encarreguen de la lectura de senyals digitals que reben, mentre que els de sortida són els que la envien. Alguns d'aquests pins tenen funcions especials dels quals cal destacar els pins sèrie RX i TX (0 i 1), que permeten la recepció i transmissió de dades, i els sis pins PWM (3, 5, 6, 9, 10 i 11), esmentats prèviament, els quals envien una senyal PWM.

Els pins analògics, a diferència dels digitals, només poden ser d'entrada. Aquests s'encarreguen de llegir les senyals analògiques enviats pels elements connectats.

És clar, totes aquestes senyals han d'arribar a un element capaç de processar tota la informació, la unitat central de processament (UCP), qui interpreta totes aquestes dades i executa unes instruccions prèviament definides.

- Pins d'alimentació (VCC) i terra (GND)

Els pins VCC són els pols positius que s'encarreguen de proporcionar un voltatge regulat als elements connectats. En el cas de l'Arduino UNO Rev3, té dos pins; un que proporciona un voltatge de 3.3V i un altre de 5V.

Els pins GND, en canvi, són els pols negatius, els quals funcionen a 0V. Bàsicament, és on acaba la corrent elèctrica.

- **Memòries Flash, SRAM i EEPROM**

Aquesta placa d'Arduino consta de tres memòries, la memòria Flash, SRAM i EEPROM.

La memòria SRAM (memòria RAM estàtica) és la zona on es creen i es manipulen les variables del programa quan s'executa. És capaç de mantenir les dades sempre i quan estigui alimentada, sense necessitat de reiniciar el circuit.

La memòria EEPROM (memòria ROM programable i suprimible elèctricament) és una memòria no volàtil, és a dir, que no varia, que permet mantenir unes dades després d'un reinici. Es pot gravar des del programa de l'Arduino.

La memòria Flash equival a un disc dur d'un ordinador. És la memòria del programa on es guarda el programa compilat.

- **Port USB**

La placa Arduino UNO Rev3 utilitza un port USB del tipus B per connectar-se a un ordinador. Des d'aquest ordinador podrem carregar un programa a la placa i, a més a més, podrem subministrar-li alimentació elèctrica.

- **Port d'alimentació**

Un cop la nostra placa tingui el seu programa guardat a la memòria, ja no caldrà connectar-la a un ordinador, però seguirà requerint alimentació. Per això mateix, aquesta placa disposa d'un port d'alimentació a on es podrà connectar diferents tipus de fonts que proporcionen alimentació.

1.2.3. Mòdul Bluetooth HC-06

Un dels elements més importants d'aquest sistema domòtic és el mòdul Bluetooth HC-06, el qual permet el control de tot el sistema amb un dispositiu mòbil.

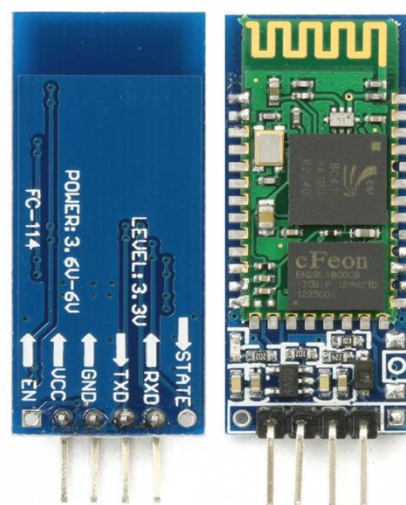
Un mòdul Bluetooth és un component compost per un receptor de radio (*CSR BC417143*), que permet la transmissió de senyals, i un processador, que s'encarrega de "traduir" les senyals rebudes i executar certes instruccions. Per poder connectar-lo a l'Arduino, inclou quatre pins: d'una banda tenim els pins TX i RX, amb els quals entrarem més a fons, d'altra tenim els pins VCC i GND, els quals es connecten als seus respectius pins de l'Arduino. Ja que el mòdul Bluetooth treballa amb un voltatge d'entre 3.3V i 6V, l'alimentarem amb el pin VCC de 3.3V de l'Arduino (tot i que es pot connectar perfectament amb 5V).

- Pin TX

El pin TX, com ja es va explicar abans, s'encarrega de la transmissió de dades. Quan es connecta al pin RX de l'Arduino, aquest rebirà les dades que ha transmet el mòdul Bluetooth, enviades prèviament des d'un dispositiu connectat a aquest.

- Pin RX

El pin RX, el qual es connecta amb el pin TX de l'Arduino, fa pràcticament l'invers que l'anterior. Aquest s'encarrega de rebre dades i les envia al dispositiu connectat, normalment per emetre un *feedback*.



**IL·LUSTRACIÓ 8. MÒDUL
BLUETOOTH HC-06**

ESPECIFICACIONS TÈCNIQUES DEL MÒDUL BLUETOOTH HC-06

Protocol Bluetooth	Bluetooth 2.0+EDR
Freqüència d'operació	2.4 GHz, banda ISM
Mode d'operació	Esclau
Modulació	GFSK (<i>Gaussian Frequency Shift Keying</i>)
Potència de transmissió	≤ 6 dBm
Sensibilitat	≤ -80 dBm
Velocitat de transmissió	Asincrònica 2.1 Mbps/120 kbps; sincrònica 1 Mbps/1 Mbps
Consum de corrent	30 mA – 40 mA
Voltatge d'operació	3.3 V – 6 V
Temperatura d'operació	-25 °C a +75 °C
Abast	5 m – 10 m
Dimensions	1.7 cm x 4 cm

**Nota: cal tenir en compte que, a l'hora de carregar un programa a l'Arduino, el pin RX ha d'estar lliure, ja que és qui s'encarrega de rebre la informació.*

1.2.4. Sensors

Sensor de presència PIR (HC-SR501)

Penso que és important prioritzar una bona seguretat per a un habitatge i, per això mateix, instal·lar una alarma dins d'aquest sistema domòtic ha de ser imprescindible. Per fer una alarma, és necessari un sensor de presència per detectar qualsevol intrusió.

Tot i que hi ha molts tipus de detectors de moviment, en aquest projecte utilitzarem un sensor infraroig passiu (PIR, *Passive Infrared*).

El sensor de presència és un dispositiu electrònic que envia senyals elèctriques quan detecta un moviment. Un sensor del tipus PIR, concretament, funciona detectant la calor emesa de les persones en forma de radiació infraroja.



**IL·LUSTRACIÓ 9. SENSOR DE PRESÈNCIA PIR
HC-SR501**

El hardware que forma aquest tipus de sensor consta principalment d'una unitat emissora i una altra receptora, de manera que pugui rebre la radiació infraroja i, seguidament, emetre unes senyals elèctriques que s'enviaran a l'Arduino. La part receptora té una zona sensible a aquesta radiació i una lent que augmenta la zona de detecció.

Respecte a les connexions, el sensor PIR té tres pins; un VCC, un GND i un de sortida, els quals es connectaran al pin d'alimentació de 5V, al pin GND i al pin digital de l'Arduino, respectivament.

ESPECIFICACIONS TÈCNIQUES DEL SENSOR DE PRESÈNCIA PIR HC-SR501

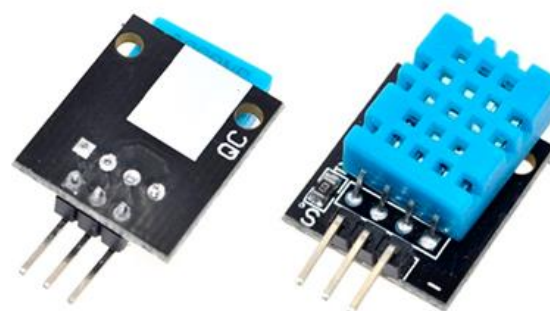
Voltatge d'operació	DC 4.5 V – 20 V
Consum de corrent estàtic	< 50 μ A
Voltatge de sortida	3.3 V (HIGH) / 0 V (LOW)
Temps de retard	5 s – 300 s
Rang de detecció	3 m – 7 m (ajustable)
Angle de detecció	< 110 °
Dimensions	32 mm x 24 mm

Sensor de temperatura i humitat (DHT11)

En una casa automatitzada estaria bé implementar un sistema de reg automàtic que s'activi quan la humitat arribi a un cert percentatge, o fins i tot que l'aire condicionat sigui capaç d'enviar aire calent o fred segons la temperatura de l'ambient. Per realitzar aquestes funcions, es requereix un sensor de temperatura i humitat.

El sensor DHT (*Digital Humidity and Temperature*) és un dispositiu de baix cost que, tal com indica el seu nom, permet mesurar la temperatura i humitat del medi exterior. Aquest sensor consta d'un capacitiu d'humitat i un termistor que mesuren l'aire, i envia les dades mesurades mitjançant una senyal digital a l'Arduino.

Tot i que aquest component té un funcionament bastant senzill, requereix una bona sincronització a l'hora de mesurar les dades. Degut a això, cal definir un marge de com a mínim dos segons, amb la finalitat d'evitar un col·lapse.



IL·LUSTRACIÓ 10. SENSOR DHT11

Cal tenir en compte que aquest sensor no és del tot precís, sinó que té un marge d'error de $\pm 2.0\text{ }^{\circ}\text{C}$ en el cas de la temperatura (que té un rang de mesura de $0\text{ }^{\circ}\text{C}$ a $50\text{ }^{\circ}\text{C}$), i un $\pm 5\%$ per la humitat (amb un rang de mesura de 20% a 90%).

De la mateixa manera que el sensor PIR, prèviament explicat, té tres pins, aquest sensor té exactament el mateix tipus de connexions. Un pin VCC, que s'alimenta també amb 5V, un pin GND que es connecta al pin terra i el pin de dades, que es connecta a un pin analògic.

ESPECIFICACIONS TÈCNIQUES DEL SENSOR DE TEMPERATURA I HUMITAT DHT11

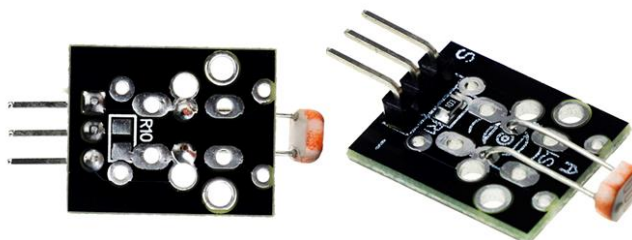
Voltatge d'operació	3.3 V – 5.5 V
Consum de corrent màxim	2.5 mA
Rang de mesurament de temperatura	$0\text{ }^{\circ}\text{C}$ – $50\text{ }^{\circ}\text{C}$
Precisió de mesurament de temperatura	$\pm 2.0\text{ }^{\circ}\text{C}$
Rang de mesurament d'humitat	20% – 90%
Precisió de mesurament d'humitat	$\pm 5\%$

Sensor de llum (KY-018)

Una altra funció que introduiré en la meva maqueta serà un sistema d'il·luminació, també automàtic, on un LED transmetrà certa intensitat de llum segons la lluminositat exterior. En aquest cas utilitzaré un sensor de llum.

El sensor de llum és un dispositiu que consta principalment d'un fotoresistor, és a dir, una resistència que varia segons la lluminositat que detecta. Llavors, quan el nivell de la llum exterior és baixa, el fotoresistor pot arribar a proporcionar fins a $1\text{ M}\Omega$, mentre que quan és alta, la resistència pot baixar fins a $0\ \Omega$. El valor d'aquesta resistència s'envia a l'Arduino mitjançant una senyal analògica, i aquesta mateixa senyal determina el valor d'intensitat de llum que ha d'emetre un LED.

Llavors, depenent de la llum exterior, el LED emetrà una intensitat de llum inversament proporcional, és a dir, quant més alt sigui el nivell de la llum exterior, menys intensitat emetrà, i quant més baix, més.



IL·LUSTRACIÓ 11. SENSOR DE LLUM KY-018

Com sempre, aquest sensor també té el pin VCC, alimentat amb 5V, i el pin GND que es connecta al pin terra. L'única novetat que té aquest element respecte als altres, és que utilitza una connexió analògica, per tant, el pin de s'encarrega d'enviar dades es connectarà a un pin analògic de l'Arduino.

ESPECIFICACIONS TÈCNIQUES DEL SENSOR DE LLUM KY-018

Voltatge d'operació	3.3V – 5V (DC)
Consum de corrent	< 0.5 mAh
Variació de resistència	$0\ \Omega$ – $1\text{ M}\Omega$
Dimensions	32 mm x 18 mm

1.2.5. Actuadors

Díode emissor de llum

Un díode emissor de llum, conegut també com LED (de les seves sigles en anglès, *Light-Emitting Diode*), és un dispositiu electrònic constituït per un material semiconductor especial, el qual és capaç de convertir una corrent elèctrica de baix voltatge en llum.

La estructura principal d'un díode consta de dos terminals (o pins), anomenats ànode (pol positiu) i càtode (pol negatiu), i un xip semiconductor, estructurada de manera que la corrent elèctrica entra per l'ànode, arriba al xip semiconductor, el qual transformarà la corrent elèctrica en llum i, finalment, surt pel càtode.



IL·LUSTRACIÓ 12. LED

És important saber que, generalment, la caiguda de tensió que té un LED és d'entre 1,8 V i 3,3 V, depenent del color del díode i la composició de metalls. En el cas d'utilitzar-lo amb un Arduino, caldrà connectar-lo amb una resistència.

ESPECIFICACIONS TÈCNIQUES DEL DÍODE EMISSOR DE LLUM (BLANC)

Voltatge d'operació	1.8 V – 3.3 V
Consum de corrent contínua	30 mA
Temperatura d'operació	-40 °C – 85 °C
Temperatura de soldadura (5 s)	260 °C
Dimensions	5 mm x 6.7 mm

Servomotor (SG-90)

Un servomotor és un tipus de motor especial que permet controlar la posició del seu eix en un moment definit.

Dins la seva caixa es situa un motor de corrent continua, que dóna mobilitat al servo, uns engranatges reductors, que converteixen part de la velocitat en força de rotació, i un circuit de control, el qual s'encarrega del control de posició del motor (rep una senyal modulada per ample de polsos (senyal PWM) i ubica el motor a la seva nova posició, depenent dels polsos rebuts).



IL·LUSTRACIÓ 13.
SERVOMOTOR SG90

Cal esmentar que existeixen dos tipus de servomotors, classificats segons les seves característiques de rotació: el d'un rang de gir limitat i el de gir continu. Tal com indiquen els seus noms, el servomotor de gir limitat només pot assolir una rotació de fins a 180°, mentre que el de gir continu és capaç de fer una rotació sencera de 360°. En aquest projecte, concretament, es farà servir un servomotor de gir continu, el SG-90.

ESPECIFICACIONS TÈCNIQUES DEL SERVOMOTOR SG-90

Voltatge d'operació	4.8 V – 6 V
Parell motor	2.5 kg/cm
Velocitat d'operació	0.1s/60°
Rotació	0° – 180°
Pes	14.7 g
Dimensions	23 mm x 12 mm x 32 mm

Brunzidor

Un bronzidor és un aparell electroacústic que produeix un brunzit continu, i sol ser utilitzat com un mecanisme d'avís o senyalització. De la mateixa manera que un LED pot convertir una corrent elèctrica en llum, aquest és capaç de convertir-la en so.

Consta principalment de dos elements, un disc piezoelèctric i una làmina metàl·lica. Quan la corrent elèctrica passa per aquest disc es produeix una ressonància elèctrica, creant d'aquesta manera sons que s'amplifiquen mitjançant la làmina metàl·lica.



IL·LUSTRACIÓ 14. BRUNZIDOR

Hi ha dos tipus de bronzidors, l'actiu i el passiu. L'actiu genera un to d'una freqüència ja predeterminada quan se li envia una senyal digital. El passiu, en canvi, pot generar tons d'una freqüència variable mitjançant una senyal PWM.

Respecte a la polaritat d'aquest component, és semblant a la d'un LED. Té un pol positiu i un altre negatiu, el pin més llarg és el pol positiu.

ESPECIFICACIONS TÈCNIQUES DEL BRUNZIDOR

Voltatge d'operació	4 V – 8 V
Consum de corrent	$\leq 30\text{mA}$
Sortida de so (10 cm)	$\geq 85\text{dB}$
Freqüència de ressonància	2300 Hz ($\pm 300\text{Hz}$)
To	Continu
Temperatura d'operació	-25 °C – 80 °C
Pes	2 g

1.3. Software

Tots aquests elements prèviament esmentats, el hardware del sistema domòtic, no poden ser controlats per l'usuari de manera espontània, sinó que requereixen l'ús del software.

Entenem com a software aquella part no tangible de la informàtica, és a dir, un programa o un conjunt de programes que inclouen dades, procediments i pautes que permeten realitzar diferents tasques en un sistema informàtic; obtenir, processar i enviar informació digital.

1.3.1. Tipus de software

El software podria classificar-se segons diverses categories, en funció del seu cost (*freeware*, *shareware*, *adware*, lliure o de pagament) o les seves finalitats (ofimàtic, empresarial, de comunicació, de seguretat, maliciós, d'oci o educatiu). Però, a nivell pràctic, el software es divideix en tres tipus generals.

- Software de sistema

El software de sistema és el conjunt de programes que gestionen el processament intern d'informació, oferint una interfície d'usuari que permet interactuar amb un sistema, controlar el hardware, i donar suport a altres programes.

- Software de programació

El software de programació fa referència al conjunt d'eines i utilitats que permeten als programadors desenvolupar programes informàtics mitjançant llenguatges de programació.

- Software d'aplicació

El software d'aplicació engloba a aquells programes dissenyats amb una interfície visual per facilitar a l'usuari la realització de tasques específiques.

1.3.2. Què és programar?

Sabem que un programa inclou dades, procediments i pautes que permeten realitzar diferents tasques en un sistema informàtic. Aquests programes, però, no sorgeixen del no res, sinó que algú els ha de programar.

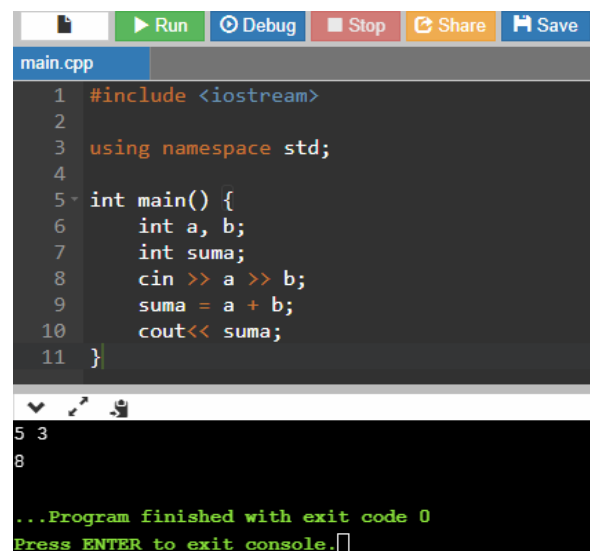
Diem que la programació, a nivell bàsic, es tracta de donar certes instruccions a un ordinador amb la finalitat de que realitzi una tasca determinada.

Ara bé, l'ordinador només pot executar instruccions escrites amb el llenguatge binari, on només s'utilitzen dos símbols, el 0 i l'1. Escriure directament en codi binari és una tasca molt complexa i, per tal d'evitar-la, es fan servir llenguatges de programació. Aquests s'utilitzen per facilitar la creació de programes capaços d'executar un conjunt determinat d'instruccions, ja que tenen un vocabulari més proper a la llengua humana. Això sí, com la llengua humana, tots aquests llenguatges de programació tenen un conjunt de regles sintàctiques i semàntiques que ha de respectar el programador a l'hora de crear el programa.

Tot i així, aquest programa segueix utilitzant un llenguatge que l'ordinador no pot llegir. Per això mateix, necessitem compilar-lo, és a dir, traduir-lo a llenguatge binari. Després d'aquest procés, es crea l'arxiu d'un programa guarda les instruccions en codi binari. Segons el llenguatge de programació utilitzat, s'ha de fer servir diferents compiladors.

Avui dia existeixen molts llenguatges de programació, i cadascun té les seves pròpies característiques. En aquest treball em centraré en el llenguatge C++, el qual s'utilitza a la plataforma d'Arduino.

A la il·lustració 15 es mostra un programa bàsic escrit en llenguatge C++ que consisteix en sumar dues variables de tipus enter.



```
main.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int a, b;
7     int suma;
8     cin >> a >> b;
9     suma = a + b;
10    cout<< suma;
11 }
```

```
5 3
8
...Program finished with exit code 0
Press ENTER to exit console.
```

IL·LUSTRACIÓ 15. PROGRAMA C++

1.3.3. Arduino IDE

L'entorn integrat de desenvolupament (IDE, *Integrated Development Environment*) es tracta d'un software de programació que, com ja s'ha explicat abans, facilita el desenvolupament de programes informàtics. Normalment estan formats per un editor de text, eines de construcció automàtiques, que automatitzen o fan més eficient la creació del programa (com la compilació del codi font a codi binari), i una eina de depuració, que identifica i corregeix errors de programació.

L'Arduino IDE és una aplicació que permet escriure i pujar un programa a la nostra placa d'Arduino, basant-se en els llenguatges de programació C i C++. Aquest entorn pot ser ampliat mitjançant l'ús de biblioteques informàtiques, les quals proporcionen funcionalitats extres al programa. S'ha de tenir en compte que el codi escrit requereix dues funcions bàsiques, una que s'executa quan s'inicia el sketch i una altra que es repeteix en bucle.

La il·lustració que es mostra a la dreta es tracta d'un exemple de programa anomenat Blink, i consisteix en una sèrie de funcions que fan que un LED connectat al pin 2 de l'Arduino s'engegui i s'apagui una i altra vegada, en bucle.

En aquest treball s'utilitzarà funcions més complexes, les quals s'explicaran a l'apartat del bloc pràctic.

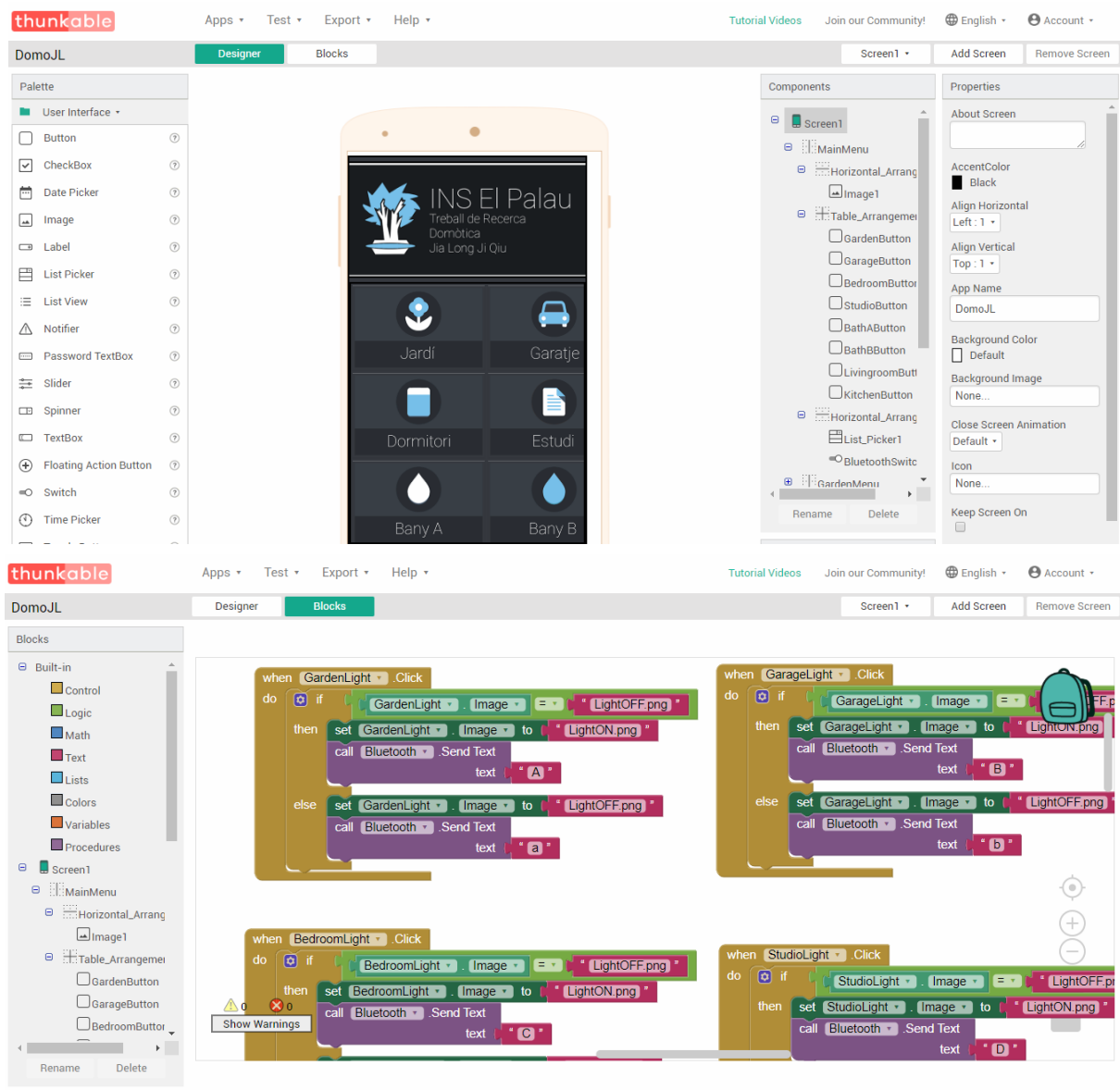
The image is a screenshot of the Arduino IDE interface. At the top, the window title is "Blink Arduino 1.8.6". Below the title bar is a menu bar with "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Underneath the menu bar is a toolbar with icons for opening files, saving, uploading, and downloading. The main text area contains the C++ code for the Blink sketch. The code includes comments in Catalan explaining the purpose of the code and the functions used. The code defines a constant LED pin number, sets up the pin mode in the setup function, and then uses the loop function to toggle the LED state (HIGH and LOW) with a 1000ms delay between each state change. The status bar at the bottom indicates "21" and "Arduino/Genuino Uno en COM3".

```
// La funció 'int' defineix un LED a un pin de l'Arduino,
// en aquest cas al pin 2.
int LED = 2;
// La funció de 'setup' executa una vegada que es prem
// la restauració o s'activa la placa
void setup() {
  // Inicialitza el pin digital 'LED' com una sortida
  pinMode(LED, OUTPUT);
}
// la funció del bucle s'executa una i altra vegada, en bucle
void loop() {
  // Encèn el 'LED' (HIGH és el nivell de voltatge)
  digitalWrite(LED, HIGH);
  // Espera d'un segon (1000ms)
  delay(1000);
  // Apaga el 'LED' amb el LOW
  digitalWrite(LED, LOW);
  // Espera d'un segon (1000ms)
  delay(1000);
}
```

IL·LUSTRACIÓ 16. EXEMPLE D'UN PROGRAMA D'ARDUINO

1.3.4. App Inventor

L'App Inventor es tracta d'un altre entorn de desenvolupament de software, concretament d'aplicacions destinades al sistema operatiu Android. Aquest entorn es divideix principalment en dues parts, un dissenyador i un editor de blocs.



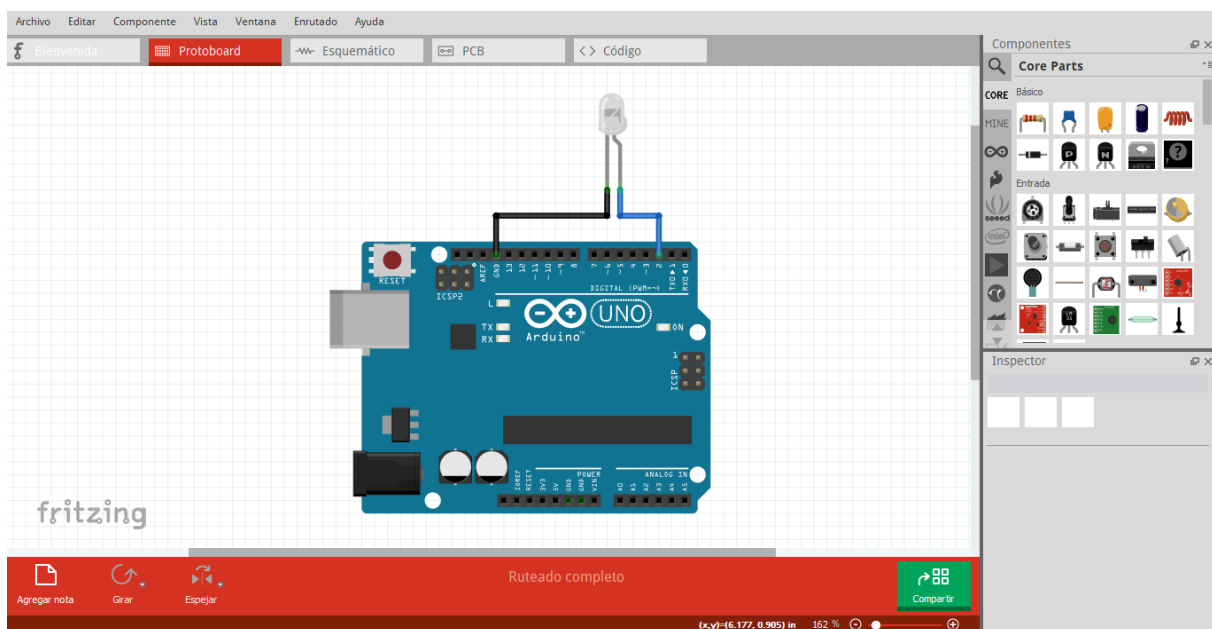
IL·LUSTRACIÓ 17. INTERFÍCIE DE L'APP INVENTOR (THUNKABLE)

La interfície del dissenyador consta de quatre apartats: la paleta, la qual ofereix diferents components (botons, textos, imatges...) que es poden afegir a l'aplicació; les propietats, que permeten modificar les característiques de cada component (així com els colors, la mida...); l'apartat dels components, on es pot seleccionar cadascun d'aquests per modificar-los; i el visualitzador, un simulador on es pot observar gràficament com és l'aplicació que s'està desenvolupant.

L'editor de blocs és la part que permet programar cada component de l'aplicació, però, en comptes d'utilitzar un llenguatge de programació escrit en text, s'utilitzen blocs per tal de representar més visualment cada funció i facilitar a l'usuari el seu ús. Aquest consta de dos apartats: un de blocs, on es pot seleccionar i afegir al programa diferents blocs que fan una funció determinada, i un visualitzador, on es mostren tots els blocs afegits.

1.3.5. Fritzing

El Fritzing és un programa que permet representar esquemàticament un circuit elèctric, inclòs aquells que funcionen amb Arduino. En el meu cas em centraré més en l'apartat de protoboard i l'esquemàtic, tot i que ofereix altres funcions com construir un circuit imprès (PCB), o programar directament des d'aquesta aplicació.



IL·LUSTRACIÓ 18. INTERFÍCIE DEL FRITZING

Les diferències que hi ha entre l'apartat de protoboard i l'esquemàtic és que al de protoboard es mostra el circuit elèctric d'una manera més gràfica, amb il·lustracions que s'assemblen més a la realitat, mentre que a l'esquemàtic s'utilitza una simbologia específica.

En tots dos casos, trobem a la dreta el menú de components, on es pot trobar els diferents elements que es poden utilitzar en un circuit.

2. BLOC PRÀCTIC

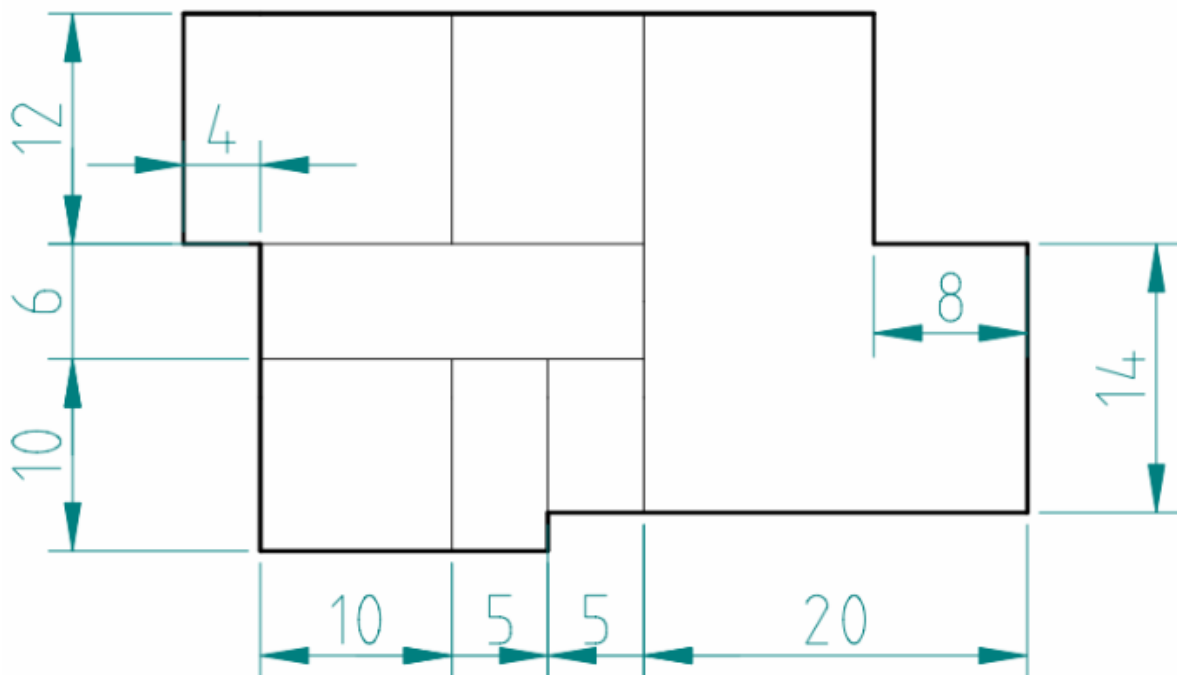
2.1. Planificació de la maqueta

2.1.1. Plànols

Abans de començar amb la construcció de la maqueta, és necessari dissenyar-la, per tant, s'ha de pensar en una idea general que es representarà amb un petit croquis, el qual determinarà la estructura principal de la maqueta, tot indicant les mesures.

En el meu cas, com que he enfocat aquest projecte sobretot a la programació, he dissenyat una maqueta bastant simple, la qual s'estructura en un jardí, un garatge, un dormitori, un estudi, dos banys, una sala d'estar i una cuina.

El material que he fet servir a l'hora de construir la maqueta és el cartó ploma, un material molt còmode per elaborar maquetes, ja que és un molt lleuger i de gran rigidesa, a més de que és molt fàcil de manipular.



IL·LUSTRACIÓ 19. PLÀNOL DE LA MAQUETA

2.1.2. Circuit elèctric

Un cop tingut clar la estructura de la nostra maqueta, només queda afegir-li el sistema domòtic, que en aquest cas es tracta d'un circuit elèctric controlat per la placa Arduino UNO Rev3.

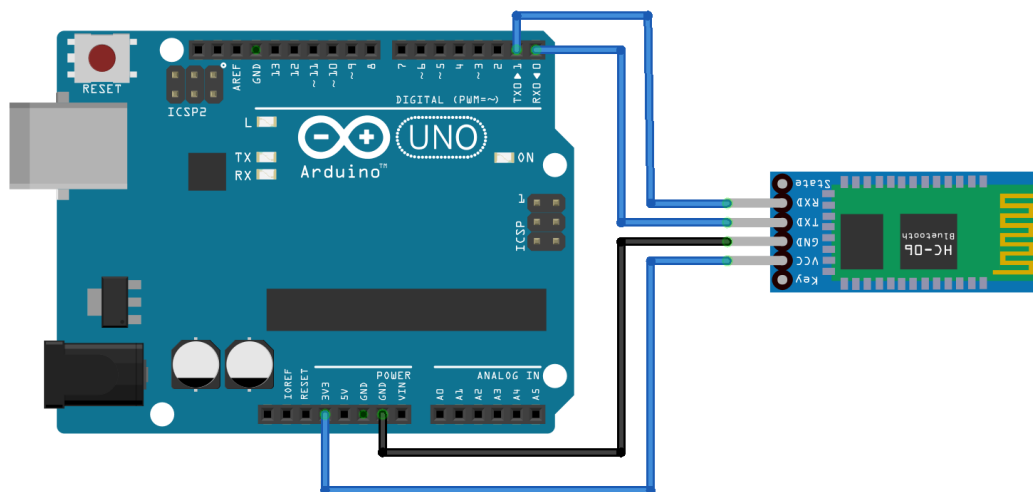
Tenint en compte les funcions que es volen implementar a cada part de la casa, s'ha d'afegir els diferents elements que realitzaran aquestes determinades funcions.

D'acord al que he vist necessari personalment, he afegit a la meua maqueta llums per a cada habitació, un sistema d'alarma, un aparell de mesura de temperatura i humitat, un sistema de llum automàtic i un mecanisme que obrirà o tancarà la porta del garatge.

A continuació, adjuntaré diferents il·lustracions que mostraran gràficament com s'ha de connectar cada component a l'Arduino.

- Mòdul Bluetooth

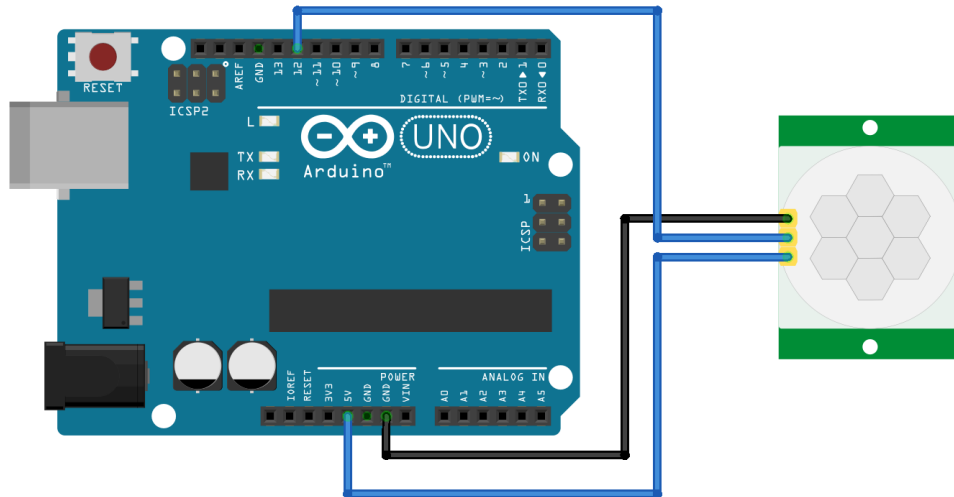
Tal com es comentava a l'apartat de hardware, el mòdul Bluetooth HC-06 té quatre pins, RX, TXD, GND i VCC. El pin RX es connecta amb el pin TX de l'Arduino, i el pin TX amb el pin RX, de manera que es pugui establir una transmissió de dades entre aquest component i la placa. El pin de voltatge, VCC, es connecta al pin 3.3V de l'Arduino, ja que pot treballar perfectament dins d'un rang de 3.3 V a 6 V, i el pin GND amb el pin GND.



IL·LUSTRACIÓ 20. CONNEXIÓ DEL MÒDUL BLUETOOTH

- Sensor PIR

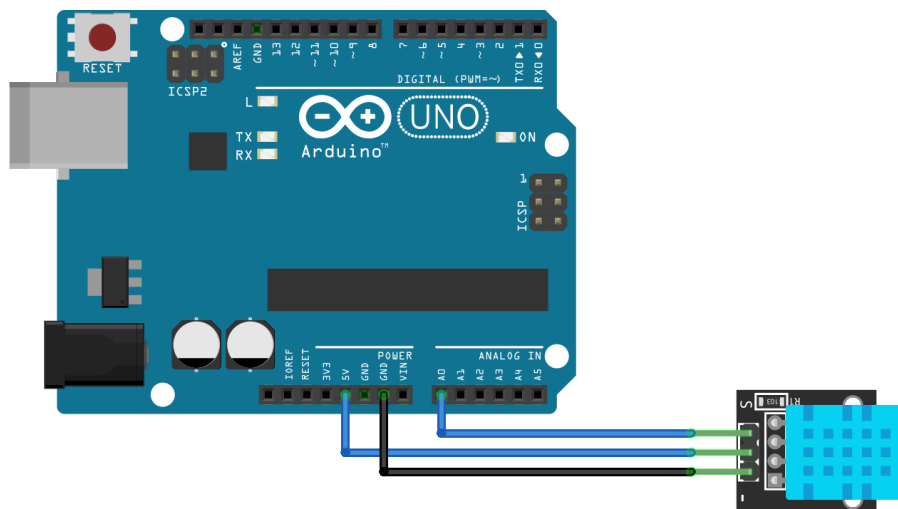
El sensor PIR té tres pins, GND, senyal i VCC. El pin senyal s'ha de connectar a un pin digital d'Arduino, de manera que quan es detecti un moviment, envii a l'Arduino una senyal digital HIGH (5V). El pin GND, com sempre, amb el pin terra, i el VCC a 5V, ja que el seu voltatge d'operació es de 4.5 V a 20 V.



IL·LUSTRACIÓ 21. CONNEXIÓ DEL SENSOR PIR

- Sensor de temperatura i humitat

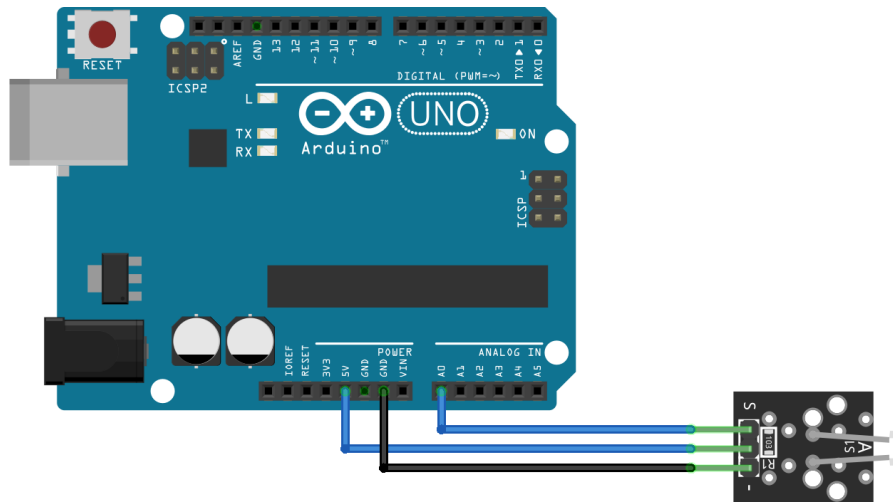
El sensor de temperatura i humitat també té els tres pins de GND, VCC i senyal. Tenint en compte que el voltatge d'operació és d'entre 3.3V a 5.5V, és indiferent connectar-lo al pin de 3.3V o 5V de l'Arduino. El pin GND es connecta a terra i la senyal, en aquest cas, es connecta a un pin analògic, ja que envia senyals analògiques per indicar els diferents valors de la temperatura i la humitat.



IL·LUSTRACIÓ 22. CONNEXIÓ DEL SENSOR DE TEMPERATURA I HUMITAT

- Sensor de llum

El sensor de llum, el qual consta també de tres pins, GND, VCC i de senyal, té unes connexions pràcticament iguals al sensor de temperatura i humitat. El seu voltatge d'operació és d'entre 3.3V a 5V, per tant, no importa a quin pin d'alimentació es connecti. El pin GND, com sempre, a terra i, per últim, el pin de senyal es connecta a un pin analògic, ja que envia una senyal analògica la qual indicarà el valor del resistor que està incorporat en aquest component.



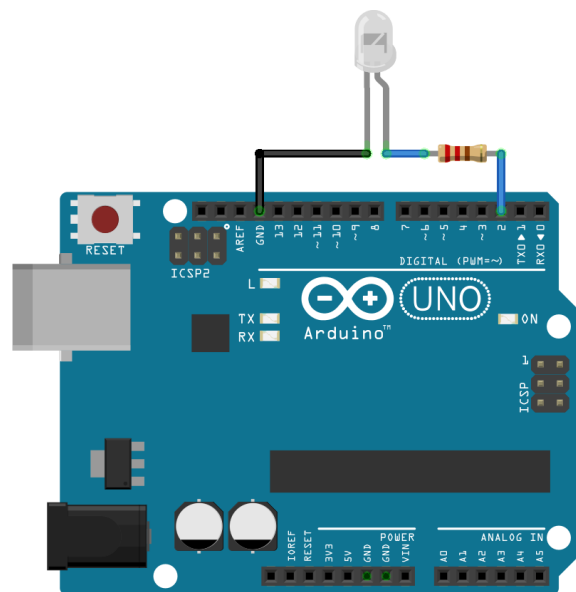
IL·LUSTRACIÓ 23. CONNEXIÓ DEL SENSOR DE LLUM

- LED

Sabent que un LED té dos terminals, el càtode (-) i l'ànode (+), cal connectar-lo amb la placa d'Arduino de manera que estigui el pol positiu connectat al pin digital corresponent, i el pol negatiu acabi en un pin GND, tot mitjançant cables.

Cal esmentar també que és necessari afegir una resistència de com a mínim 200Ω, per tal d'evitar que es cremi.

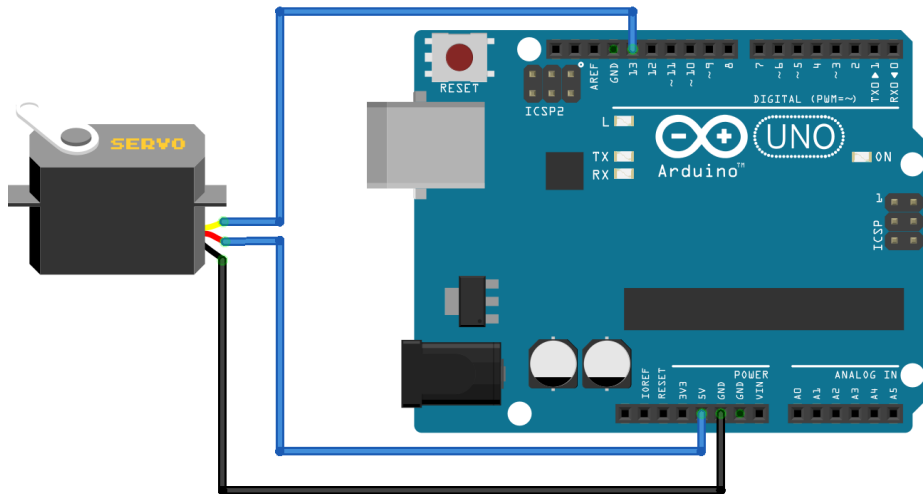
**Nota: per connectar tots els càtodes amb el pin GND, es pot unir tot amb un únic cable.*



IL·LUSTRACIÓ 24. CONNEXIÓ DEL LED

- Servomotor

El servomotor té tres pins, GND, VCC i senyal. Aquest actuator, com que necessita una alimentació contínua, s'ha de connectar al pin d'alimentació de 5V, ja que el seu voltatge d'operació és d'entre 4.8V a 6V. El pin de senyal es connecta a un pin digital i el de GND a terra.

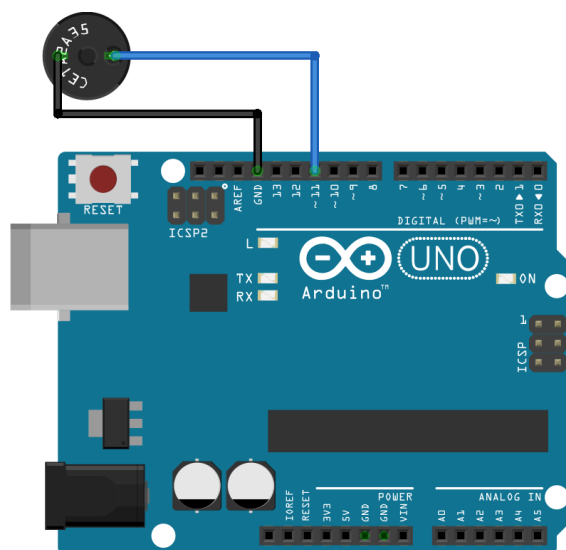


IL-LUSTRACIÓ 25. CONNEXIÓ DEL SERVOMOTOR

- Brunzidor

Per últim, tenim el brunzidor, que simplement es connecta el pol positiu amb el pin digital corresponent, i el pol negatiu amb el pin GND de la placa.

Aquest component, a diferència del LED, no necessita cap resistor, ja que el seu voltatge d'operació és de 4V a 8V.



IL-LUSTRACIÓ 26. CONNEXIÓ DEL BRUNZIDOR

2.2. Programació

2.2.1. Programa d'Arduino

En aquest apartat explicaré les diferents funcions que utilitzo en el meu programa d'Arduino.

Biblioteques informàtiques

Les biblioteques informàtiques proporcionen funcionalitats extres a un programa, és a dir, són un conjunt de funcions que tenen unes finalitats úniques i faciliten molt la feina de l'usuari a l'hora de programar.

En el programa s'utilitzen tres biblioteques, les del servomotor, del sensor de presència PIR i del sensor de temperatura i humitat.

```
#include <Servo.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

Declaració de variables

Una variable és una manera de nombrar i emmagatzemar un valor de manera que pugui ser utilitzat i manipulat posteriorment pel programa, com per exemple les dades que recullen els sensors.

Declarar una variable vol dir definir el seu tipus i, opcionalment, un valor inicial. És recomanable declarar-les fora de la funció set-up i loop.

A un programa d'Arduino normalment a l'hora de declarar les variables se li declaren un valor inicial que indicarà al pin al que es connecta.

La declaració de variables és molt pràctic perquè si es vol canviar la connexió d'un actuator o un sensor a un altre pin, és tan senzill com canviar el número del pin assignat a aquest element, en comptes d'haver de canviar-lo en tot el programa. A més a més, permet identificar més fàcilment els valors que s'utilitzen, ja que es defineixen els noms dels elements que es connecten a cada pin de l'Arduino.

En aquest programa he declarat, mitjançant una funció int, la majoria de variables com els LEDs i els sensors als pins que els pertoca. D'altra banda tenim altres variables com el 'char valor', 'int alarma', 'int moviment' o 'int valorres = 0', valors que aniran canviant en la funció loop, els quals s'explicaran posteriorment. Finalment, tenim altres funcions com el 'DHT dht(A0,DHT11)', procedent de la llibreria '<DHT.h>', que permet declarar fàcilment aquest

sensor al seu pin corresponent, i el 'Servo servo', que li dona un nom al servomotor que utilitzaré a la funció set-up.

```
// Variables del sensor DHT11
DHT dht(A0,DHT11);

// Variables dels LEDs
int ledgaratge = 2;
int leddormitori = 4;
int ledestudi = 5;
int ledbanya = 6;
int ledbanyb = 7;
int ledsala = 8;
int ledcuina = 9;

// Variables del sensor de llum
int ledjardi = 3;
int resistencia = A1;
int valorres = 0;

// Variables de l'alarma
int ledpassadis = 10;
int buzz = 11;
int PIR = 12;
int alarma;
int moviment;

// Variable del servomotor
Servo servo;

char valor;
```

Funcions que es realitzen una vegada (set-up)

La majoria d'aquestes funcions que es realitzen una vegada s'inicia el programa.

La funció 'pinMode', la més utilitzada, consisteix en definir els pins com entrada (input), de manera que la placa Arduino rebrà senyals, o sortida (output), on serà la placa qui enviarà les senyals. En aquest cas, els pins dels sensors s'han de definir com entrada, ja que l'Arduino rep les senyals que envien aquests, mentre que els dels actuadors són de sortida, ja que reben senyals.

També he utilitzat altres funcions com ‘Serial.begin(9600)’, la qual estableix la velocitat de transmissió en bits per segon per la transmissió de dades en sèrie. En aquest cas, la placa Arduino s’ha de comunicar amb el mòdul Bluetooth, el qual utilitza els pins RX i TX, en una velocitat predeterminada de 9600 bits per segon.

La funció ‘servo.attach(13)’, procedent de la llibreria ‘<Servo.h>’, connecta la variable servo al pin assignat, en aquest cas al pin 13.

```
//Funcions que es realitzen un cop,  
//quan el programa s'inicia  
void setup() {  
    Serial.begin(9600);  
    pinMode(ledjardi, OUTPUT);  
    pinMode(ledgaratge, OUTPUT);  
    pinMode(leddormitori, OUTPUT);  
    pinMode(ledestudi, OUTPUT);  
    pinMode(ledbanya, OUTPUT);  
    pinMode(ledbanyb, OUTPUT);  
    pinMode(ledsala, OUTPUT);  
    pinMode(ledcuina, OUTPUT);  
    pinMode(ledpassadis, OUTPUT);  
    pinMode(PIR, INPUT);  
    pinMode(buzz, OUTPUT);  
    servo.attach(13);  
}
```

Funcions que es repeteixen (*loop*)

Abans de començar amb la explicació de les funcions utilitzades dins del *loop*, aquelles que es repeteixen indefinidament, cal que aclareixi la funció base d’aquest sistema domòtic.

Com que és un sistema que funciona amb Bluetooth, controlat des d’un mòbil, l’usuari ha d’enviar d’alguna manera una senyal que després l’Arduino ha de reconèixer per realitzar certa funció. En el meu cas, el funcionament base del programa consisteix en enviar lletres. Des de l’aplicació mòbil, el qual s’explicarà en l’apartat següent, l’usuari ha de prémer un botó i, segons de quin sigui, enviarà una lletra diferent al Bluetooth connectat al mòbil prèviament, el mòdul Bluetooth, connectat a l’Arduino, rebrà i enviarà aquesta senyal a la placa i, segons la lletra que sigui, farà una funció concreta. Per exemple, prémer un botó per encendre un LED enviarà al Bluetooth la lletra ‘A’, quan l’Arduino llegeixi la senyal que enviï el Bluetooth

reconeixerà aquesta lletra i realitzarà la funció assignada a la lletra enviada, que en aquest cas consistirà en enviar una senyal de 5V al pin del LED corresponent.

La lectura de lletres es fa amb la funció 'if(Serial.available() > 0) valor = Serial.read()', la qual comprova primerament si hi ha un dispositiu, el mòdul Bluetooth en aquest cas, connectat als pins sèrie, i si es compleix definirà la variable 'valor' amb 'Serial.read()', és a dir, el valor que llegeixi del mòdul Bluetooth, el qual serà una lletra. Aquest procés, ja que està dins del bucle, es repetirà contínuament i, cada vegada que s'envii una lletra diferent, el valor de la variable també canviarà.

```
//Funcions que es repetiran
void loop() {
  if(Serial.available() > 0 ) valor = Serial.read();
```

El LED del jardí varia la seva il·luminació depenent de la llum exterior, un mecanisme que consta del sensor de llum. Quan l'Arduino rebí la lletra 'A' (en aquest cas), l'Arduino llegirà les senyals enviades per aquest sensor a través del pin analògic que se li ha assignat. Aquesta senyal depèn del valor de la resistència d'un fotoresistor, el qual varia segons la llum que detecta, i aquest valor pot variar entre 0 i 1023. Com que el valor màxim que es pot enviar a un pin PWM és 255, cal fer una conversió mitjançant regla de tres de manera que el rang de 0 a 1023 canviï a un rang de 0 a 255, proporcionalment. D'aquesta manera, al LED se li assignarà un valor que determinarà la intensitat de la llum emesa, on 0 és la mínima i 255 la màxima.

```
//LED automàtic
if(valor == 'A') {
  valorres = analogRead(resistencia);
  valorres = valorres * 255 / 1023;
  analogWrite(ledjardi,valorres);
}
if(valor == 'a') digitalWrite(ledjardi,LOW);
```

La resta de LEDs no tenen molt misteri, quan l'Arduino rebí una determinada lletra majúscula, enviarà una senyal de 5V al pin corresponent de manera que l'encendrà i, quan rebí una lletra minúscula, enviarà una senyal de 0V, la qual l'apagarà.

```
//LEDs
//LED del garatge
if(valor == 'B') digitalWrite(ledgaratge,HIGH);
if(valor == 'b') digitalWrite(ledgaratge,LOW);
```


Tenint en compte que la porta del garatge variarà amb un angle de 90°, quan l'Arduino rebi la lletra 'S' realitzarà la funció 'servo.write(90)', procedent de la llibreria <Servo.h>, que bàsicament envia una senyal al pin que se li ha assignat al servomotor de manera girarà un angle definit amb aquesta mateixa funció, en aquest cas els 90°. Quan rebi la 's' minúscula tornarà a la seva posició inicial de 0° amb 'servo.write(0)'.

```
//Porta del garatge
if(valor == 'S') servo.write(90);
if(valor == 's') servo.write(0);
```

En relació amb la temperatura i la humitat, com que es tracta simplement d'un sistema domòtic simulat dins d'una maqueta, vaig decidir que s'indiquessin tots dos paràmetres a través de la interfície de l'aplicació mòbil.

Dins del programa es defineix abans amb una declaració de tipus 'float', el qual permet obtenir xifres amb decimals, a les variables de la 'temperatura' i la 'humitat'. Aquests valors varien contínuament utilitzant les funcions de lectura 'dht.readTemperature()' i 'dht.readHumidity()', ambdues funcions procedents de la llibreria '<DHT.h>', les quals permeten a l'Arduino llegir les senyals analògiques que envia el sensor de temperatura i humitat. Un cop definides aquestes variables, es realitzen les funcions 'Serial.print()' que permetran enviar al mòdul Bluetooth, a través del pin de transmissió TX, els paràmetres que apareixeran posteriorment a l'aplicació mòbil.

```
//Temperatura
float temperatura;
float humitat;
temperatura = dht.readTemperature();
humitat = dht.readHumidity();
Serial.print(temperatura);
Serial.print(" °C");
Serial.print("|");
Serial.print(humitat);
Serial.print(" %");
Serial.print("|");
delay(2000);
```

Per simular una alarma he utilitzat el sensor de presència i un brunzidor, i aprofitant aquest sensor també he afegit un LED automàtic. Dins del programa, aquesta funció es divideix en dues parts, una d'aquestes consisteix en encendre un LED quan es detecta moviment, l'altra és gairebé igual, però en comptes d'encendre un LED s'encén un brunzidor.

La variable 'moviment' varia entre 'HIGH' (5V) o 'LOW' (0V), és a dir, o detecta moviment o no, i per això s'utilitza la funció 'digitalRead(PIR)', ja que es tracta d'una senyal digital. Com que quan s'activa una alarma s'ha de mantenir activat fins que es desactivi manualment, he declarat la variable 'alarma' que serà 0 o 1, depenent de la lletra rebuda, si és 'Z' serà '1', i si és 'z' serà '0' i, a més a més, apagarà el bronzidor. El que s'aconsegueix amb aquesta funció és mantenir un valor constant, de manera que en un inici, com que fora del loop ja s'havia declarat la variable 'alarma' com '0', el que farà l'Arduino serà encendre un LED durant 5 segons quan es detecti moviment, és a dir, quan sigui 'HIGH'. Però, quan rebi la lletra 'Z' aquesta variable canviarà de ser '0' a '1', i mentre aquest valor segueixi sent 1, en comptes d'encendre un LED encendrà un bronzidor el qual emetrà un so constant que només podrà desactivar-se a través de l'aplicació.

```
//Alarma
moviment = digitalRead(PIR);
if(valor == 'Z') alarma = 1;
if(valor == 'z') {
    alarma = 0;
    digitalWrite(buzz, LOW);
}
if(alarma == 0) {
    if(moviment == HIGH) {
        digitalWrite(ledpassadis, HIGH);
        delay(5000);
        digitalWrite(ledpassadis, LOW);
    }
}
if(alarma == 1) {
    if(moviment == HIGH) {
        digitalWrite(buzz, HIGH);
    }
}
```

2.2.2. Aplicació mòbil

Interfície d'usuari

A l'hora de dissenyar la interfície d'usuari de l'aplicació mòbil vaig tenir en compte uns quants factors importants, tots pensats per beneficiar l'usuari. Vaig pensar primerament que la interfície havia de ser senzilla, sense cap sobrecàrrega d'informació, amb la finalitat d'obtenir comoditat a l'hora d'utilitzar l'aplicació. També havia de ser intuïtiva, de manera que l'usuari, sense cap coneixement previ sobre aquesta interfície, fos capaç d'utilitzar-la sense cap mena d'explicació. Finalment em vaig fixar en la estètica, la qual engloba el conjunt d'icones, textos i els colors, on tenia pensat una paleta de colors semblant a un blau fosc, de manera que la interfície no cansi molt els ulls, textos amb una font de lletra simple i unes icones capaces de representar gràficament les seves funcions.

Passant ara a la part pràctica d'aquesta aplicació mòbil, vaig idear una estructura principal que consistia en un menú principal, on es mostren els botons de les diferents parts de la casa, la connexió de Bluetooth i informació extra, i uns menús secundaris, on es mostren els botons que controlen els actuadors que es troben en la seva respectiva part.



És clar, tenint en compte les idees i els objectius que volia complir amb aquesta aplicació, vaig canviar tota aquesta interfície manualment, és a dir, modificar els colors, afegir imatges, canviar l'estil de text, etc. Finalment, el resultat va ser el següent:

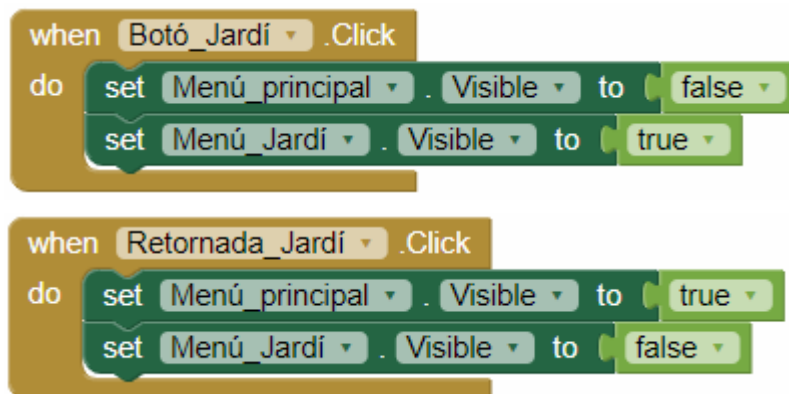


Tota aquesta estructura, però, no funciona automàticament, sinó que els botons han de ser programats per realitzar la funció desitjada, ja sigui canviar d'un menú a altre, connectar-se a Bluetooth, enviar informació al mòdul Bluetooth, etc.

Programació per blocs

La plataforma de l'App Inventor utilitza la programació per blocs, la qual és molt semblant a la programació per text però aquest és més visual i entenedor, enfocat sobretot per aquelles persones que no estan molt familiaritzades en aquest àmbit.

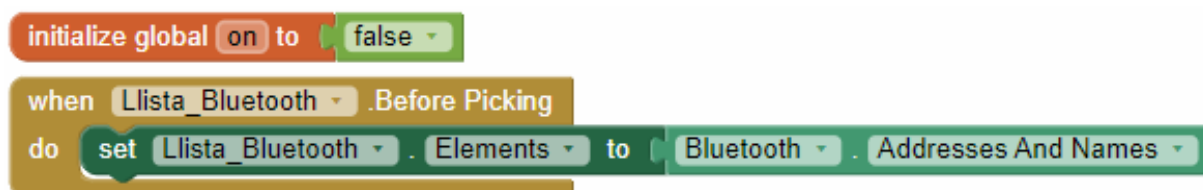
El primer conjunt de blocs consisteix en el canvi de menús, del principal al secundari i viceversa. Per fer aquests canvis, vaig utilitzar la funció d'establir la visibilitat d'un menú, de manera que quan es fa clic a un botó concret, un menú es farà invisible i, simultàniament, un altre serà visible. El mateix procés s'aplica als altres botons que fan la mateixa funció.



Per gestionar la connexió Bluetooth vaig pensar en un sistema bastant complex. El botó que permet connectar i desconnectar el Bluetooth és un interruptor, de manera que quan sigui clicat apareix una llista dels dispositius Bluetooth vinculats al mòbil, i per desconnectar-lo s'havia de tornar a clicar.

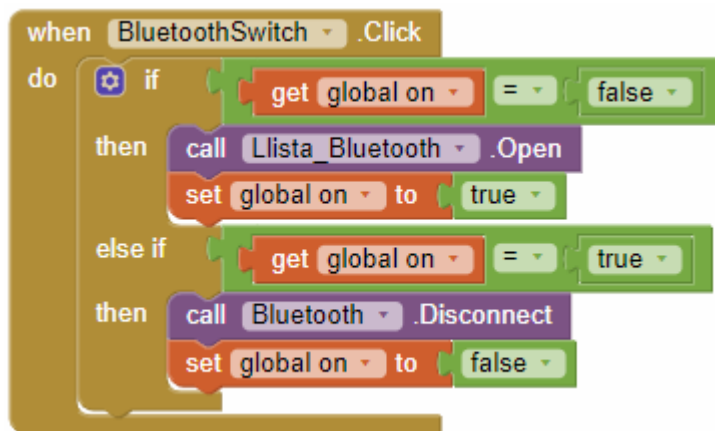
Per programar aquest sistema, cal definir abans els dos estats de l'interruptor, en el meu cas vaig declarar una variable amb el nom 'on', el qual s'inicia amb el valor 'false', ja que suposem que quan s'obre l'aplicació, aquest interruptor està apagat. El bloc següent estableix que els elements d'una llista com els noms dels dispositius Bluetooth vinculats al dispositiu mòbil.

Aquestes dues funcions s'inicien una vegada s'obre l'aplicació, ara bé, per accedir a aquesta llista de dispositius Bluetooth i escollir a quin connectar-se, cal afegir dos blocs més. Un d'aquests blocs permet que, amb un únic interruptor, es pugui obrir la llista dels dispositius Bluetooth o desconnectar la connexió Bluetooth.

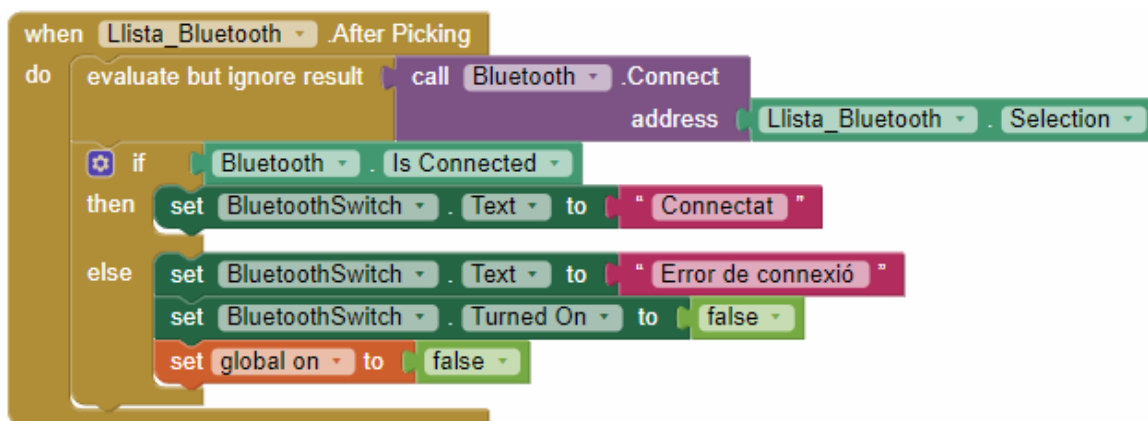


Les funcions d'aquest bloc, en un llenguatge més entenedor, estableixen que, quan l'interruptor sigui clicat, si la variable 'on' és 'false', és a dir, si l'interruptor està apagat, obrirà la llista de dispositius Bluetooth i canviarà el valor de la variable 'on' a 'true', ja que l'interruptor s'activarà. Si aquesta primera condició no es compleix i la variable 'on' és 'true', és a dir, si l'interruptor està activat, quan sigui clicat desconnectarà la connexió Bluetooth i canviarà la variable 'on' a 'false'. D'aquesta manera ja hem establert les dues funcions de l'interruptor.

El bloc restant s'encarrega de comprovar si s'ha pogut establir la connexió Bluetooth correctament, d'aquesta manera, s'envia un feedback a l'usuari fent-li saber si els dispositius s'han connectat o no.



Després d'escollir un element de la llista, el dispositiu Bluetooth al qual es vol connectar, el dispositiu mòbil es connectarà mitjançant Bluetooth a aquesta selecció. Un cop realitzat aquesta funció, es comprova si el Bluetooth està connectat: si s'ha connectat correctament, el text "Connecta't a Bluetooth" que apareix inicialment al costat de l'interruptor s'establirà com "Connecta't", en cas contrari, si no s'ha pogut connectar, s'establirà aquest mateix text com "Error de connexió" i, a més a més, canviarà l'estat de l'interruptor a apagat i el valor de la variable 'on' a 'false'.



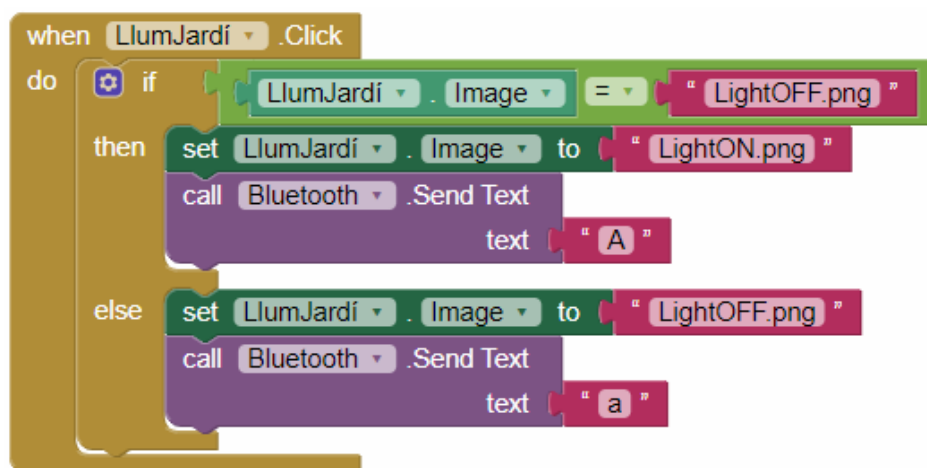
Vet aquí ara les funcions dels botons que enviarn les lletres al dispositiu Bluetooth connectat, les quals l'Arduino rebrà i seguidament executarà unes determinades instruccions.

Aquests blocs són pràcticament idèntics, ja que la finalitat de tots aquests botons són enviar lletres que seran interpretades per l'Arduino, qui farà la resta. Tot i així, em va semblar important afegir una funció extra, destinada també a donar un feedback a l'usuari, la qual consisteix en un canvi d'imatge. Per explicar-lo d'una manera més pràctica, quan es dona clic a un botó per engegar un llum, la imatge inicial que representa que el llum està apagat canviarà a una altra, simulant que s'ha engegat:



Aquesta és, doncs, la funció de cada botó que s'encarrega d'enviar les lletres per engegar els llums. Quan aquest botó es clicat, si la seva imatge representa que el llum està apagat, canviarà a una altra la qual representa que està engegat, enviant també una lletra majúscula al dispositiu Bluetooth al qual està connectat, en el cas de l'exemple la lletra "A". Si aquesta condició no es compleix, és a dir, si el llum està engegat, canviarà a la imatge anterior i enviarà la lletra minúscula "a".

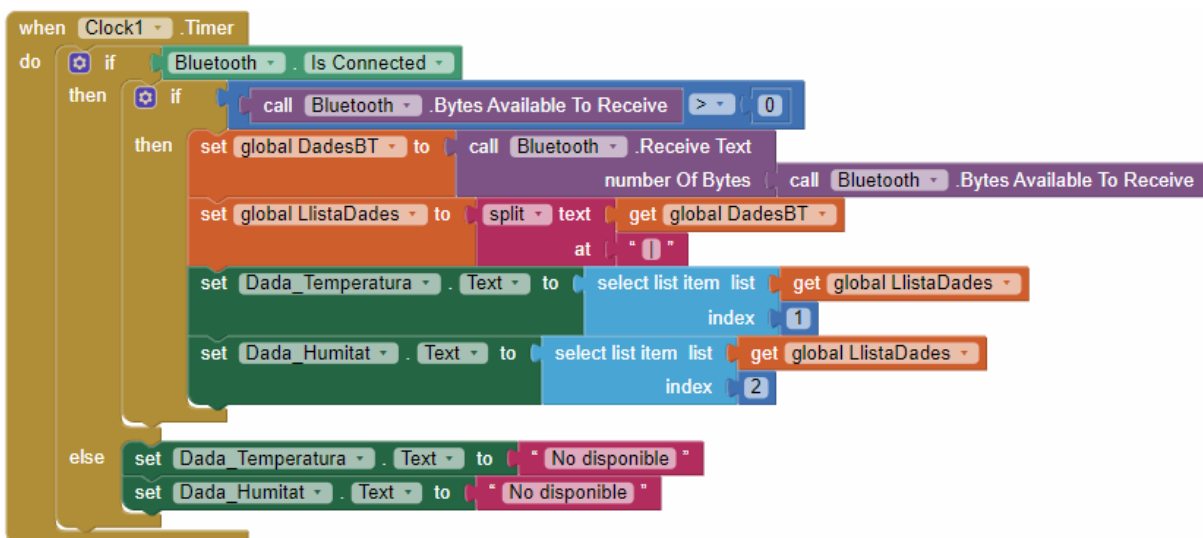
En el cas dels botons encarregats d'administrar la porta de garatge i l'alarma, també s'utilitzen les mateixes funcions, però enviant lletres diferents. És clar, les imatges d'aquests botons són diferents.



Finalment resta l'últim conjunt de blocs, el qual s'encarrega d'indicar la temperatura i la humitat de l'ambient. Els primers blocs consisteixen en funcions que es realitzen quan s'obre l'aplicació, les quals declaren variables. En aquest cas, es declara la variable 'DadesBT' sense cap valor, la qual canviarà contínuament el seu valor definit per les dades de la temperatura i la humitat, i també la variable 'LlistaDades', que es tracta d'una llista que recull els valors definits de la variable 'DadesBT'.



La resta de blocs s'encarreguen d'interpretar aquestes dades, mitjançant funcions que es repeteixen indeterminadament. Primer es comprova si el Bluetooth està connectat, si es així, si es reben dades del dispositiu Bluetooth connectat, en forma de bits (un sistema de numeració binari), s'establirà la variable 'DadesBT' com els bits rebuts del Bluetooth. Com que el sensor de temperatura i humitat envia més d'una dada alhora (el valor de la temperatura, el símbol "°C", la humitat i el símbol "%"), s'ha d'afegir un símbol que l'aplicació reconegui com un separador (en aquest cas, he utilitzat el símbol "|" com separador, el qual envia l'Arduino entre els valors corresponents de la temperatura i la humitat), de manera que es puguin utilitzar les dues dades independentment. Un cop dividides, s'afegeixen ambdues dades a la llista de dades prèviament esmentada i, seguidament, s'estableix el text que indica el valor de la temperatura com el primer element de la llista i l'altre text que indica el de la humitat com el segon element (ja que aquest sistema està programat de manera que la primera dada enviada sigui el valor de la temperatura i, seguidament, la de la humitat. En el cas de que el Bluetooth no estigui connectat o es desconnecti, ambdós textos s'establiran com "No disponible".



3. CONCLUSIONS

En un principi, aquest treball em va suposar un gran repte degut a la manca de formació personal dins d'un àmbit gairebé desconegut. Tot i així, després d'una àmplia recerca sobre la domòtica, la electricitat, l'Arduino i la programació, vaig aconseguir complir el meu objectiu principal.

Referent a la part teòrica, durant l'elaboració d'aquesta vaig aprendre molt més del que m'esperava. D'una banda, tot i que no m'atreia del tot el tema d'automatització d'habitatges, em va semblar bastant interessant sobretot pel funcionament d'aquesta, el qual em va resultar més simple del que em pensava, arribant finalment a la conclusió de què la domòtica és un sistema molt innovador que permet facilitar molt la vida diària de les persones. D'altra, parlant ara de l'Arduino, els diferents components utilitzats i la programació, vaig aprendre funcionaments totalment nous tant dels components com de la placa de l'Arduino UNO R3. Tot i que a nivell teòric ja estava bastant familiaritzat amb aquests conceptes, durant la recerca he pogut reforçar molt més aquests coneixements.

Pel que fa a la part pràctica, he de dir que la construcció de la maqueta ha sigut molt entretinguda. Encara havent-me causat un munt de problemes a l'hora de l'elaboració d'aquesta, ja que a vegades no aconseguia els resultats que m'esperava, els resultats finals van complir totalment les meves expectatives. El desenvolupament de l'aplicació mòbil ha sigut sense dubte la part més divertida; el fet d'haver d'enfocar-la a les necessitats de l'usuari, ideant una interfície que s'adeqüi a aquestes, a més d'afegir les funcions necessàries mitjançant la programació, em va semblar un desafiament realment apassionant. Tot i així, cal dir que tant la maqueta com l'aplicació mòbil estan en un procés de millora i ampliacions que, malgrat sense disposar d'uns recursos professionals, s'han intentat elaborar de la millor manera possible.

En definitiva, aquest treball ha sigut sense dubte una experiència molt enriquidora que m'ha ajudat sobretot a ampliar els meus coneixements sobre la programació, que era el que més m'interessava, a més d'haver-me permès aplicar els meus coneixements d'electrònica en un entorn més proper a un àmbit professional.

4. WEBGRAFIA

- </tipos de>. (2018). *Tipos de hardware*. Recollit de <https://www.tiposde.org/informatica/145-tipos-de-hardware/>
- </tiposde>. (2018). *Significado de software*. Recollit de <https://www.tiposde.org/informatica/938-significado-de-software/>
- 40deFiebre. (2018). *¿Qué debe tener una buena interfaz de usuario?* Recollit de <https://www.40defiebre.com/elementos-buena-interfaz-usuario>
- Aprendiendo Arduino. (2018). *Memoria Flash, SRAM y EEPROM*. Recollit de <https://aprendiendoarduino.wordpress.com/2017/06/21/memoria-flash-sram-y-eeprom-3/>
- Aprendiendo Arduino. (2018). *Qué es Arduino*. Recollit de <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- Aprendiendo Arduino. (2018). *Qué es Arduino y el Hardware Libre*. Recollit de <https://aprendiendoarduino.wordpress.com/2015/03/22/que-es-el-hardware-libre/>
- Arduino. (2018). *Arduino - PWM*. Recollit de <https://www.arduino.cc/en/Tutorial/PWM>
- Arduino. (2018). *Arduino Uno Rev3*. Recollit de <https://store.arduino.cc/arduino-uno-rev3>
- Arduino para todos. (2018). *Módulo Bluetooth HC-06 con Arduino y Android*. Recollit de <http://arduparatodos.blogspot.com/2017/06/modulo-bluetooth-hc-06-con-arduino-y.html>
- Arduino.cl. (2018). *¿Qué es Arduino?* Recollit de <http://arduino.cl/que-es-arduino/>
- ARQCompus. (2018). *Domótica*. Recollit de <http://arqcompus-domotica.blogspot.com/>
- Domodesk. (2018). *¿Qué es Domótica?* Recollit de <https://www.domodesk.com/126-que-es-domotica.html>
- DomoPrac. (2018). *El Protocolo de comunicaciones, el lenguaje de la domótica*. Recollit de <http://www.domoprac.com/protocolos-de-comunicacion-y-sistemas-domoticos/el-protocolo-de-comunicaciones-el-lenguaje-de-la-domotica.html>
- DomoPrac. (2018). *Sensores: Tipos y Funcionalidades*. Recollit de <http://www.domoprac.com/hardware-y-productos-domoticos/sensores-tipos-y-funcionalidades.html>
- instructables. (2018). *PIR Motion Sensor Tutorial*. Recollit de <https://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/>
- IONOS. (2018). *UI: ¿Qué es una interfaz de usuario?* Recollit de <https://www.ionos.es/digitalguide/paginas-web/disenio-web/ui-que-es-una-interfaz-de-usuario/>
- JOY. (2018). *Tipologia de redes: Domotica*. Recollit de <http://tipologiasdredes.blogspot.com/2015/07/domotica.html>
- Llamas, L. (2018). *Controlar un servo con Arduino*. Recollit de <https://www.luisllamas.es/controlar-un-servo-con-arduino/>
- Llamas, L. (2018). *Salidas analógicas PWM en Arduino*. Recollit de <https://www.luisllamas.es/salidas-analogicas-pwm-en-arduino/>

- Lupion, F. (2018). *PROYECTOS DE TECNOLOGÍA: El buzzer o zumbador*. Recollit de <https://elblogdelprofesordetecnologia.blogspot.com/2016/02/el-buzzer-o-zumbador.html>
- Minitronica. (2018). *Uso del sensor de movimiento HC-SR501 con Arduino*. Recollit de <https://www.minitronica.com/uso-del-sensor-movimiento-hc-sr501-arduino/>
- National Instruments. (2018). *¿Que es una Señal Modulada por Ancho de Pulso (PWM) y Para Qué es Utilizada?* Recollit de <http://digital.ni.com/public.nsf/allkb/AA1BDEA4AA224E3E86257CE400707527>
- Naylamps Mechatronics. (2018). *Tutorial Básico de Uso del Módulo Bluetooth HC-06 y HC-05*. Recollit de https://www.naylampmechatronics.com/blog/12_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html
- NEO. (2018). *Medios de transmisión*. Recollit de <http://neo.lcc.uma.es/evirtual/cdd/tutorial/fisico/Mtransm.html>
- Omniblug. (2018). *Artículo (Sensor de temperatura y humedad DHT1 y DHT22)*. Recollit de <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>
- Programarfacil. (2018). *Cómo utilizar el DHT11 para medir la temperatura y la humedad con Arduino*. Recollit de <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>
- TechKrowd. (2018). *Arduino: Buzzer pasivo (KY-006 37in1 kit)*. Recollit de <https://www.youtube.com/watch?v=9rInAEhHTkI>
- TechKrowd. (2018). *Arduino: Cómo configurar un módulo bluetooth HC-06 (Slave)*. Recollit de <https://www.youtube.com/watch?v=M7DTEcdzDTI&t=745s>
- TechKrowd. (2018). *Arduino: Fotorresistencia (KY-018 37in1 kit)*. Recollit de <https://www.youtube.com/watch?v=ApbHmCpT3qc&t=671s>
- TechKrowd. (2018). *Arduino: Sensor temperatura y humedad DHT11 (KY-015 37in1 kit)*. Recollit de <https://www.youtube.com/watch?v=sxtP16dAVno&t=387s>
- Velneo. (2018). *Software de programación*. Recollit de <https://velneo.es/software-de-programacion/>
- Vendomótica. (2018). *Definición de controlador domótico*. Recollit de <http://vendomotica.com/blog/definicion-de-controlador-domotico/>

ANNEXOS

5. ANNEXOS

5.1. Annex 1, programa de l'Arduino

```
#include <Servo.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

// Variables del sensor DHT11
DHT dht(A0,DHT11);

// Variables dels LEDs
int ledgaratge = 2;
int leddormitori = 4;
int ledestudi = 5;
int ledbanya = 6;
int ledbanyb = 7;
int ledsala = 8;
int ledcuina = 9;

// Variables del sensor de llum
int ledjardi = 3;
int resistencia = A1;
int valorres = 0;

// Variables de l'alarma
int ledpassadis = 10;
int buzz = 11;
int PIR = 12;
int alarma = 0;
int moviment;

// Variable del servomotor
Servo servo;

char valor;
```

```
//Funcions que es realitzen un cop,  
//quan el programa s'inicia  
void setup() {  
    Serial.begin(9600);  
    pinMode(ledjardi,OUTPUT);  
    pinMode(ledgaratge,OUTPUT);  
    pinMode(leddormitori,OUTPUT);  
    pinMode(ledestudi,OUTPUT);  
    pinMode(ledbanya,OUTPUT);  
    pinMode(ledbanyb,OUTPUT);  
    pinMode(ledsala,OUTPUT);  
    pinMode(ledcuina,OUTPUT);  
    pinMode(ledpassadis,OUTPUT);  
    pinMode(PIR,INPUT);  
    pinMode(buzz,OUTPUT);  
    servo.attach(13);  
}  
  
//Funcions que es repetiran  
void loop() {  
    if(Serial.available() > 0 ) valor = Serial.read();  
  
    //LED automàtic  
    if(valor == 'A') {  
        valorres = analogRead(resistencia);  
        valorres = valorres * 255 / 1023;  
        analogWrite(ledjardi,valorres);  
    }  
    if(valor == 'a') digitalWrite(ledjardi,LOW);  
  
    //LEDs  
    //LED del garatge  
    if(valor == 'B') digitalWrite(ledgaratge,HIGH);  
    if(valor == 'b') digitalWrite(ledgaratge,LOW);  
    //LED del dormitori  
    if(valor == 'C') digitalWrite(leddormitori,HIGH);  
    if(valor == 'c') digitalWrite(leddormitori,LOW);  
    //LED de l'estudi  
    if(valor == 'D') digitalWrite(ledestudi,HIGH);  
    if(valor == 'd') digitalWrite(ledestudi,LOW);  
    //LED del bany A  
    if(valor == 'E') digitalWrite(ledbanya,HIGH);  
    if(valor == 'e') digitalWrite(ledbanya,LOW);
```

```
//LED del bany B
if(valor == 'F') digitalWrite(ledbanyb,HIGH);
if(valor == 'f') digitalWrite(ledbanyb,LOW);
//LED de la sala d'estar
if(valor == 'G') digitalWrite(ledsala,HIGH);
if(valor == 'g') digitalWrite(ledsala,LOW);
//LED de la cuina
if(valor == 'H') digitalWrite(ledcuina,HIGH);
if(valor == 'h') digitalWrite(ledcuina,LOW);

//Porta del garatge
if(valor == 'S') servo.write(90);
if(valor == 's') servo.write(0);

//Temperatura
float temperatura;
float humitat;
temperatura = dht.readTemperature();
humitat = dht.readHumidity();
Serial.print(temperatura);
Serial.print(" °C");
Serial.print("|");
Serial.print(humitat);
Serial.print(" %");
Serial.print("|");
delay(2000);

//Alarma
moviment = digitalRead(PIR);
if(valor == 'Z') alarma = 1;
if(valor == 'z') {
    alarma = 0;
    digitalWrite(buzz,LOW);
}
if(alarma == 0) {
    if(moviment == HIGH) {
        digitalWrite(ledpassadis,HIGH);
        delay(5000);
        digitalWrite(ledpassadis,LOW);
    }
}
if(alarma == 1) {
    if(moviment == HIGH) {
        digitalWrite(buzz,HIGH);
    }
}
}
```

5.2. Annex 2, programa de l'aplicació mòbil

